Advances in real-world applications for legged robots — Source link

C. Dario Bellicoso, Marko Bjelonic, Lorenz Wellhausen, Kai Holtmann ...+4 more authors

Institutions: ETH Zurich

Related papers:

- ANYmal - toward legged robots for harsh environments

- BigDog, the Rough-Terrain Quadruped Robot

- Dynamic Locomotion Through Online Nonlinear Motion Optimization for Quadrupedal Robots

- ANYmal - a highly mobile and dynamic quadrupedal robot

- Gait and Trajectory Optimization for Legged Systems Through Phase-Based End-Effector Parameterization

Advances in real-world applications for legged robots

Author(s):
Bellicoso, C. Dario; Bjelonic, Marko; Wellhausen, Lorenz (ID); Holtmann, Kai; Günther, Fabian; Tranzatto, Marco; Fankhauser, Péter; Hutter, Marco (ID)

Advances in Real-World Applications for Legged Robots[*]

C. Dario Bellicoso[†]   **Marko Bjelonic**   **Lorenz Wellhausen**   **Kai Holtmann**

Fabian Günther   **Marco Tranzatto**   **Péter Fankhauser**   **Marco Hutter**

Abstract

This paper provides insight into the application of the quadrupedal robot ANYmal in outdoor missions of industrial inspection (ARGOS Challenge) and search and rescue (European Robotics League (ERL) Emergency Robots). In both competitions, the legged robot had to autonomously and semi-autonomously navigate in real-world scenarios to complete high-level tasks such as inspection and payload delivery. In the ARGOS competition, ANYmal used a rotating LiDAR sensor to localize on the industrial site and map the terrain and obstacles around the robot. In the ERL competition, additional Real-Time Kinematic (RTK)-Global Positioning System (GPS) was used to co-localize the legged robot with respect to a Micro Aerial Vehicle (MAV) that creates maps from the aerial view. The high mobility of legged robots allows overcoming large obstacles, e.g. steps and stairs, with statically and dynamically stable gaits. Moreover, the versatile machine can adapt its posture for inspection and payload delivery. The paper concludes with insight into the general learnings from the ARGOS and ERL challenges.

1   Introduction

Deploying and using a mobile robot in an outdoor scenario is coupled with several challenges. A real-world environment is typically unstructured, unknown and can exhibit obstacles which have to be negotiated. Proper sensing capabilities (e.g. a global positioning system or a laser scanner) are required to localize the robot and map such environments. Moreover, these sensors should be properly integrated into the mobile robot's hardware, as water and dust can pose severe limitations to the reliability of the entire machine. The execution of the path planning and navigation algorithms, together with the control framework which is implemented to execute robust locomotion, can require very high computation capabilities.

Although these are very demanding requirements, recent years have seen several advances in hardware design, computation capabilities and control algorithms which enhance the autonomy and robustness of a mobile robot. Thanks to this progress, mobile robots become ready for new applications to tackle problems like industrial inspection, mining, agriculture, search and rescue missions, and other areas. These environments require robots that can cope with obstacles, rough outdoor terrain and steps. When deployed in an environment designed for humans, the robot needs to deal with stairs, doors, and other impediments.

Figure 1: The quadrupedal robot ANYmal can be employed for industrial inspection (left) and search and rescue missions (right) in harsh environments.

Instead of creating a specialized robot for every application, over the last years legged robots have proven to serve as a multi-purpose machine that can be easily adapted to suit all kind of environments. As a proof of concept, our work discusses two applications where such a robot can be potentially deployed. Fig. 1 shows the quadrupedal robot ANYmal (Hutter et al., 2016b) in use for industrial inspection on an industrial site (Hutter et al., 2018) (left) and as med-kit delivery robot in a search and rescue scenario (right).

The ARGOS Challenge [1], initiated by TOTAL SA, aims at the application of mobile robotic solutions for offshore oil and gas site inspection. According to (Kydd et al., 2015), the organizers expect a major impact with respect to *i) health, safety and environment (i.e. a reduction of risk to personnel, environment and installation* as well as *ii) operation (i.e. cost reduction, increase of efficiency and production)*. In the ARGOS Challenge, the robots must be able to autonomously navigate on the industrial site and inspect various objects such as pressure gauges, water level gauges, or valve handle positions. They need to analyze the sound of the running pumps in order to identify malfunctioning systems, detect alarm signals, find gas leaks as well as hot spots, and recognize changes that were made on the site (e.g. missing or moved objects). To make the scenario as realistic as possible, the applied robots must satisfy ATEX (explosion protection) and Ingress Protection (IP) standards. Moreover, during the missions that happen on multiple floors connected by steep stairs, the robots are facing different hurdles such as unexpected obstacles, heavy waterfalls, strong winds, or humans that are working on the site. In contrast to specific robotic devices that are already commercially used in tanks, vessels, or pipes[2] the ARGOS Challenge seeks for very generic robots that can one-to-one take over tasks performed by human specialists. In particular, the requirements regarding mobility are demanding such that the few existing solutions like the wheel-based robots MIMROex (Pfeiffer et al., 2011) or SENSABOT(JPT Staff, 2012) are not applicable. To address these issues, four of the five ARGOS Challenger teams selected in 2013 use tracked vehicles (Kydd et al., 2015), while we propose an innovative solution based on a versatile legged robot.

The initiative to push robots for search and rescue missions started in the early 90s after the Hanshin-Awajii earthquake in Kobe, Japan, and the bombing of the Murrah federal building in Oklahoma City, United States (Murphy et al., 2008; Davids, 2002). Recent accidents like the disaster at the Fukushima nuclear power plant in Japan emphasized the need for robotic solutions that can be used to assist human search and rescue teams in hazardous environments. The ERL Emergency Robots (formerly known as *euRathlon*) is an outdoor robotics competition focused on multi-domain (air, land and sea) emergency response scenarios. Inspired by the disaster in Japan, it aims to create real-world robotics challenges for outdoor robots in demanding emergency response events (Winfield et al., 2016; Röning et al., 2017).

The 2017 competition was based in Piombino, Italy. It focused on the collaboration of land, air and under-

---

[1]http://www.argos-challenge.com

[2]For example, see http://inspection-robotics.com

water robots to inspect a simulated disaster scenario and search for survivors to whom a medkit should be delivered. The environment in which the robots had to move in was a very challenging one. The path to be followed was a very rough one and featured different types of terrain, including seaside shore, sticks and high slopes, which can push the locomotion capabilities of a walking robot to the limits.

Similar to the projects above, the DARPA Robotics Challenge (DRC) aims to work up human-scaled robots for disaster response (Pratt and Manzo, 2013). In contrast to other solutions for search and rescue, the DRC had high requirements regarding mobility and demanded robots that could interface with human environments and use human tools. These high requirements created many legged-based systems like the Atlas robot (Johnson et al., 2015; Kuindersma et al., 2016; Feng et al., 2015), WALK-MAN (Tsagarakis et al., 2017), HRP-2 (Kaneko et al., 2015), DRC-Hubo (Zucker et al., 2015), and more.

The examples above focus on bipedal robots developed for complex disaster environments. Despite all the advances throughout the DRC, the locomotion performance of humanoid robots is still far behind the performance of multi-legged robots in terms of speed, energetic efficiency, and obstacle negotiation skills. Examples of multi-legged robots with greater locomotion capabilities include HyQ (Semini et al., 2011) and its successor HyQ2Max (Semini et al., 2017), MIT's Cheetah (Seok et al., 2015) and Cheetah II (Park et al., 2017), ETH's StarlETH (Hutter et al., 2012). Boston Dynamics' Spot and SpotMini robot, a direct successor to Big Dog (Raibert et al., 2008), show impressive results of which unfortunately no scientific publications are available.

In contrast to many of the above-mentioned research platforms, ANYmal has been extensively tested in real world applications and under harsh conditions. In this paper we focus on presenting the evaluation of this machine in different outdoor scenarios, namely autonomous industrial inspection and tele-operated search and rescue, together with a description of the choices made regarding system design and the planning and control algorithms which have enabled the robot to locomote autonomously and in tele-operated mode through rough and challenging environments.

2   Mechatronic Setup of ANYmal

We build upon the modular and lightweight quadrupedal robot ANYmal (Hutter et al., 2016b) as a transporter platform for inspection and as a mobile platform for search and rescue missions (Fig. 2). The legs of this versatile machine are driven by twelve equal series elastic actuator units (Hutter et al., 2016a) mounted at the joints. The robot is designed to achieve a large range of motion, allowing to overcome obstacles and stairs. The structure also allows convenient transportation, compact storage, and simple deployment by a single operator. To keep the design as lightweight as possible, most of the structure is manufactured using carbon fibers. For fall protection, the robot features a rollover bar, a Kevlar belly plate, and shock absorbers. Moreover, force sensors in the feet provide haptic feedback of the environment, enabling safe locomotion even with no exteroceptive feedback. ANYmal is designed in a hierarchical manner: On joint level, every actuator module is connected over a Controller Area Network (CAN) bus and works independently. This allows component-level ingress and ATEX protection as well as fast and simple maintenance in case of hardware failure. On system level, the software stack runs on three independent computers that are connected through an internal network. The first computer (locomotion) hosts all real-time critical elements required for locomotion control and interfaces with the joint modules. The second computer (navigation) is responsible for environmental perception, localization, navigation, and mission execution, i.e. all software parts that are required to autonomously operate the robot. The third computer (inspection) runs all algorithms for inspection and detection.
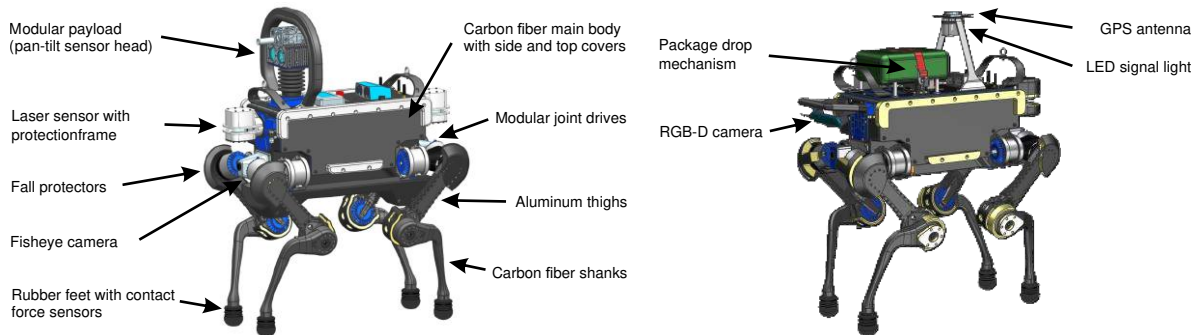
Figure 2: ANYmal can be embedded with different sets of sensors and tools depending on the scenario in which the robot is deployed. A pan-tilt sensor head (left) is used for inspection tasks. A payload drop mechanism and a GPS mounting system (right) are employed for search and rescue scenarios.

2.1 Transporter Platform for Inspection

To perceive the characteristics of its surroundings, ANYmal is equipped with two rotating Hokuyo Light Detection and Ranging (LiDAR) sensors (Hokuyo UTM-30LX-EW). Fig. 2 depicts the mounting of the two sensors. They provide detailed scans of the environment and the terrain. Using our point cloud based localization framework, the laser scans can be matched against ground truth data. This is the approach taken in the ARGOS challenge, where TOTAL provided a Computer Aided Design (CAD) model of the site to be inspected.

To execute inspection tasks, we employ a pan-tilt head containing a high-quality zoom camera with high infrared (IR) sensitivity, a thermal camera, an ultrasonic and a regular microphone, a gas detection sensor, and Light Emitting Diode (LED) illuminators. Since our robot can move its base in all directions, there is no need to employ the robot with an extension mechanism or arm.

For possible tele-operation scenarios, the robot is additionally equipped with two wide-angle cameras pointing forward and backward, providing an omnidirectional view of the robot's surroundings. Combining the camera images with the 3D laser point cloud allows us to create a virtual reconstruction of the environment and provides the operator with a 3rd person view. Additionally, these cameras are used to detect humans around the robot for safe interaction with humans.

The proposed system can operate fully autonomous with onboard batteries for more than 2.5 hours. To extend this lifetime, ANYmal can autonomously dock to recharge the battery and to pressurize the main body with Nitrogen for ATEX compliance (Kolvenbach and Hutter, 2017).

2.2 Mobile Platform for Search and Rescue Missions

Outdoor missions, such as those executed during the ERL challenge, might be executed in environments which do not have the geometric richness needed to properly make use of localization algorithms. To overcome this limitation, ANYmal can be equipped with a Piksi Multi[3] module, an RTK GPS receiver mounted on the main body of the robot. This component, in combination with a fixed RTK base station, is able to provide position measurements with an accuracy between 1.0 cm to 5.0 cm on the horizontal plane and between 1.5 cm to 8.0 cm in vertical direction. However, position measurements become unreliable when the robot is in the proximity of tall structures which interfere with the sensing capabilities of the Piksi Multi. For this reason, ANYmal is also equipped with one rotating laser sensor in the back, as described in Section 2.1. Our localization framework fuses the multiple sources of position measurements and relies more on the laser

---

[3]Piksi Multi Robot Operating System (ROS) driver available online at https://github.com/ethz-asl/ethz_piksi_ros
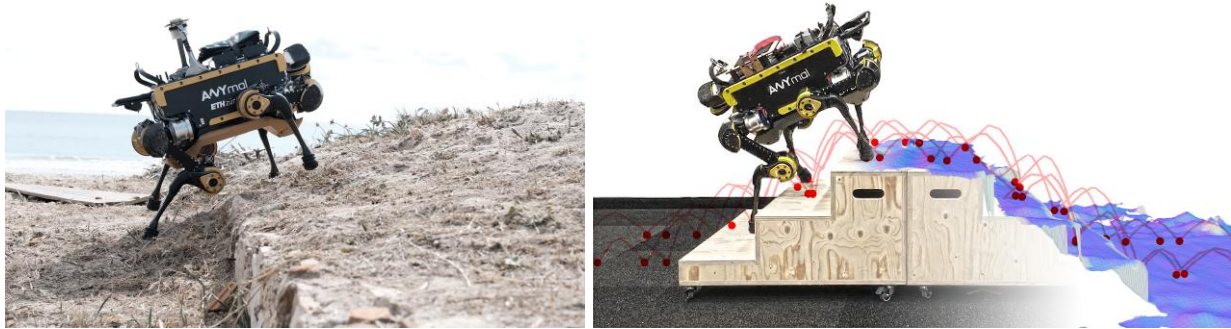
Figure 3: The robot ANYmal walking blindly on a sandy slope at the ERL challenge (left) and climbing a set of stairs (right). Blind walking is based on an optimization-based approach to controlling a dynamic gait, i.e., trotting gait. Climbing over challenging obstacles like stairs includes map-based motion planning. Moreover, footholds (red dots) and base and leg motions (red lines) are generated by processing a terrain map with the help of onboard sensors (Fankhauser et al., 2018a).

sensor when the robot is locomoting close to buildings.

To deliver a med-kit to a victim at the ERL challenge, the pan-tilt head in Section 2.1 is replaced by a custom payload delivery system. The first aid kit is strapped onto a carbon fiber plate, which is rigidly attached to the upper part of ANYmal's main body. Beneath the carbon fiber plate, a servo motor and a belt mechanism are installed. By rotating the servo, the belt mechanism is unlatched, similar to the ones used in cars as safety belts and pulled in by a rotational spring.

For navigation and foothold planning, the machine is equipped with an RGB-D camera in the front, which provides detailed scans of the environment and the terrain. During the ERL challenge, we have been working with the RealSense ZR300 (Keselman et al., 2017), which outputs high-resolution depth images at a higher update rate compared to the rotating laser sensor. The sensor is based on active stereo vision which uses a projected light pattern to improve the stereo matching algorithm. The active light projection also allows to perceive objects with homogeneous appearance and stereo matching is also possible in dark environments (Fankhauser, 2018).

3 Locomotion in Rough Terrain

Autonomous locomotion on ground that is not perfectly deterministic requires locomotion control strategies that are robust against irregularities. Moreover, larger obstacles and more challenging terrain features require a full control pipeline including terrain perception, map processing, and map-based motion planning. Fig. 3 shows both control approaches in action.

3.1 Quadrupedal Robot Locomotion through Optimization-Based Control

Several modules contribute to the robustness of locomotion in rough terrain. The following section gives an overview of terrain estimation and adaptation, motion generation and motion execution.

3.1.1 Terrain Estimation and Adaptation

Legged locomotion on high gradient slopes like in Fig. 3 (left) imposes several challenges since the robot's configuration needs to be adapted to avoid slipping and kinematic limits. Without exteroceptive sensing, the robot is only able to estimate the terrain properties by incorporating state estimation and joint measurements.

We locally model the terrain using a three dimensional plane, which is defined by a fixed point $\mathbf{r}_0 = [x_0, y_0, z_0]^T$ and a normal vector $\mathbf{n} = [a, b, c]^T$. As described in (Gehring et al., 2015), given a generic point $\mathbf{r} = [x, y, z]^T$, we model the plane by writing

$$(\mathbf{r} - \mathbf{r}_0)^T \mathbf{n} = \mathbf{0} \implies z = \frac{d - ax - by}{c}, d = ax_0 + by_0 + z_0. \tag{1}$$

Assuming that $c = 1$, i.e. the plane will never be perpendicular to the world-frame z-axis, there are three parameters which need to be estimated, namely $\boldsymbol{\pi} = [a, b, d]^T$. We solve a least-squares problem by fitting the plane through the most recent contact location $\mathbf{r}_i = [x_i, y_i, z_i]$ of each leg. The height z_i of each of these positions can be written as a linear function of the parameters $\boldsymbol{\pi}$

$$z_i = \begin{bmatrix} -x_i & -y_i & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ d \end{bmatrix}. \tag{2}$$

By collecting the four measured contact heights, we compute the estimated plane parameters $\hat{\boldsymbol{\pi}} = [\hat{a}, \hat{b}, \tilde{d}]^T$ by writing

$$\underbrace{\begin{bmatrix} \vdots \\ z_i \\ \vdots \end{bmatrix}}_{\boldsymbol{\zeta}} = \underbrace{\begin{bmatrix} \ddots & & 1 \\ -x_i & -y_i & \vdots \\ & \ddots & 1 \end{bmatrix}}_{\mathbf{H}} \begin{bmatrix} \hat{a} \\ \hat{b} \\ \hat{d} \end{bmatrix} \implies \hat{\boldsymbol{\pi}} = \mathbf{H}^\dagger \boldsymbol{\zeta}, \tag{3}$$

where $\mathbf{H}^\dagger$ is the pseudo-inverse of $\mathbf{H}$. From the estimation of these parameters we define a so called *Control Frame C* which is aligned to the estimated local terrain and to the heading direction of the robot. This enables the generation of motion plans which keep the torso of the robot always parallel to the local estimation of the terrain, i.e. this approach allows to blindly and robustly (Gehring et al., 2016) overcome challenging environments, e.g., sloped terrain.

3.1.2 Motion Generation

The locomotion behavior of ANYmal can be specified through a periodic contact schedule (or *gait pattern*), or through incoming contact events. In the former case, the gait pattern defines, in phase domain, the lift-off and touch-down timings of each leg. This information, coupled with the actual estimated contact state, defines whether a leg is sustaining the overall motion of the robot (i.e., it is a *support* leg) or it is *swinging* to a new desired contact location. Based on the specific timing of the gait pattern, different gaits can be achieved. During the ERL and the ARGOS challenges we used a trotting gait on flat and on terrain like the one depicted in Fig. 3 (left). A trot is a dynamic gait that schedules two diagonal legs to be in ground contact at each time and the other two legs to be in swing motion. This results in faster movements with respect to other static gaits and at the same time achieves robust locomotion with respect to disturbances due to the high step frequency that allows for step corrections (Gehring et al., 2015; Gehring et al., 2016). Experimental results at the ERL challenge have shown that adding a small stance phase between each swing phase lead to an increase in the overall locomotion robustness.

Several motion planning and control strategies are implemented to guarantee balance while walking. First, the location of the planned footholds is computed based on a modified version of the *inverted pendulum* model (Gehring et al., 2016), which computes footholds as a function of the torso velocity error. To define the complete swing leg trajectory of each foot, a fifth-order spline is fitted through the lift-off and the touch-down locations. Second, the torso motion is planned such that the whole-body center of mass lies, at all times, over the support polygon (i.e., the convex hull of the contact locations). The height of the main body is kept constant to a user-specified value w.r.t. the local estimation of terrain. The orientation of the estimated terrain, computed in 3.1.1, is used as a reference orientation of the main body.

3.1.3 Motion Execution

To track the torso motion reference described in 3.1.2, a *Virtual Model Controller* is implemented which generates a virtual wrench $\mathbf{w}^*$ that acts on the main body of the robot. We compute the components of this wrench as a function of the reference torso position $\mathbf{r}_{IB}^*$, velocity $\mathbf{v}_B^*$ and acceleration $\mathbf{a}_B^*$, as well as desired rotation quaternion $\mathbf{q}_{IB}^*$ and angular velocity $\boldsymbol{\omega}_{IB}^*$. The two components $\mathbf{f}^*$ and $\mathbf{t}^*$ of the wrench are given by

$$
\begin{aligned}
\mathbf{f}^* &= \mathbf{K}_p^f(\mathbf{r}_{IB}^* - \mathbf{r}_{IB}) + \mathbf{K}_i^f \int (\mathbf{r}_{IB}^* - \mathbf{r}_{IB}) + \mathbf{K}_d^f(\mathbf{v}_B^* - \mathbf{v}_B) + m\mathbf{a}_B^* - \sum_{k \in \mathcal{B}} \mathbf{f}_k^g \\
\mathbf{t}^* &= \mathbf{K}_p^t(\mathbf{q}_{IB}^* \boxminus \mathbf{q}_{IB}) + \mathbf{K}_i^t \int (\mathbf{q}_{IB}^* \boxminus \mathbf{q}_{IB}) + \mathbf{K}_d^t(\boldsymbol{\omega}_{IB}^* - \boldsymbol{\omega}_{IB}) - \sum_{k \in \mathcal{B}} \hat{\mathbf{r}}_{IS_k} \mathbf{f}_k^g,
\end{aligned}
\tag{4}
$$

where $\mathbf{K}$ are diagonal gain matrices, $\mathcal{B}$ is the set of the actuated rigid bodies, $\mathbf{f}_k^g$ is the gravitational force acting at the center of mass of the k-th body, $\hat{\mathbf{r}}_{IS_k}$ is the skew-symmetric matrix function of the position of the center of mass of body k such that $\hat{\mathbf{r}}_{IS_k} \mathbf{f}_k^g = \mathbf{r}_{IS_k} \times \mathbf{f}_k^g$, and $\boxminus$ is the box-minus operator (Bloesch et al., 2016a). The wrench integrates feed-forward terms such as gravity compensation and reference acceleration, as well as feed-back terms which encode deviation of the main body estimated pose and twist from the reference motion. At each control loop, the virtual wrench is distributed to the legs which are in contact through a *Contact Force Distribution* algorithm, a constrained optimization problem which computes reference contact forces for each leg. These are computed such that certain constraints are not violated, e.g. such that the contact forces are inside the friction cone (to avoid slipping) and such that the actuator commands are limited (i.e. to avoid torque limits). This is formulated as a Quadratic Programming problem as

$$
\boldsymbol{\lambda}^* = \operatorname{argmin} \quad (\mathbf{A}\boldsymbol{\lambda} - \mathbf{b})^T \mathbf{S}(\mathbf{A}\boldsymbol{\lambda} - \mathbf{b}) + \boldsymbol{\lambda}^T \mathbf{W} \boldsymbol{\lambda}
$$

$$
\underbrace{\begin{bmatrix} \cdots & \mathbf{I} & \cdots \\ \cdots & \hat{\mathbf{r}}_{BF_i} & \cdots \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \vdots \\ \boldsymbol{\lambda}_i \\ \vdots \end{bmatrix}}_{\boldsymbol{\lambda}} = \underbrace{\begin{bmatrix} \mathbf{f}^* \\ \mathbf{t}^* \end{bmatrix}}_{\mathbf{b}}
\tag{5}
$$

$$
\text{s. t.} \qquad
\begin{aligned}
&\lambda_{N,i} \geq \lambda_{N,min} \\
&\|\boldsymbol{\lambda}_{T,i}\| \leq \mu \lambda_{N,i} \\
&\tau_{min} \leq \tau_{j,j}(\lambda) \leq \tau_{max},
\end{aligned}
$$

where $\mathbf{S}$ is a diagonal weighting matrix used to penalize different components of the cost function, μ is an estimated friction coefficient, $\mathbf{r}_{BF_i}$ is the position vector of the i-th foot, λ_N is the normal component of the contact force, which is aligned with the terrain normal estimated in (2), and $\boldsymbol{\lambda}_T$ is the contact force projected onto the estimated terrain. Finally, assuming quasi-static dynamics (i.e. only the gravity terms from the equations of motion are considered), the actual actuator commands τ^* are computed as

$$
\tau_k^* = -\sum_{i \in \mathcal{I}(\mathbf{q})} \mathbf{J}_i^T \boldsymbol{\lambda}_i^* - \sum_{k \in \mathcal{B}} \mathbf{J}_k^T \mathbf{f}_k^g,
\tag{6}
$$

where $\mathbf{J}_k$ is the translation Jacobian of the k-th leg respectively. To track the motion of the swing legs, a joint-space motion controller is implemented which tracks joint positions and velocities obtained through inverse kinematics of Cartesian space reference states (see Fig. 4).

To decide whether a leg is part of the contact force distribution or if it should be swinging, knowledge about its contact state is required. We have implemented our control algorithms based both on force estimation and on optical force sensor measurements. In both cases, we obtain a measurement or estimation of the contact forces $\mathbf{F_{ext}} \in \mathbb{R}^{n_e}$ acting on each foot of the robot.

Force sensors can be employed to directly measure the reaction forces acting on the feet. While these provide a direct measurement, they are prone to drift in the raw data. This can be compensated by implementing outlier detection algorithms, which in turn increases the complexity of the contact detection.

(a) Main body Position

(b) Terrain estimation

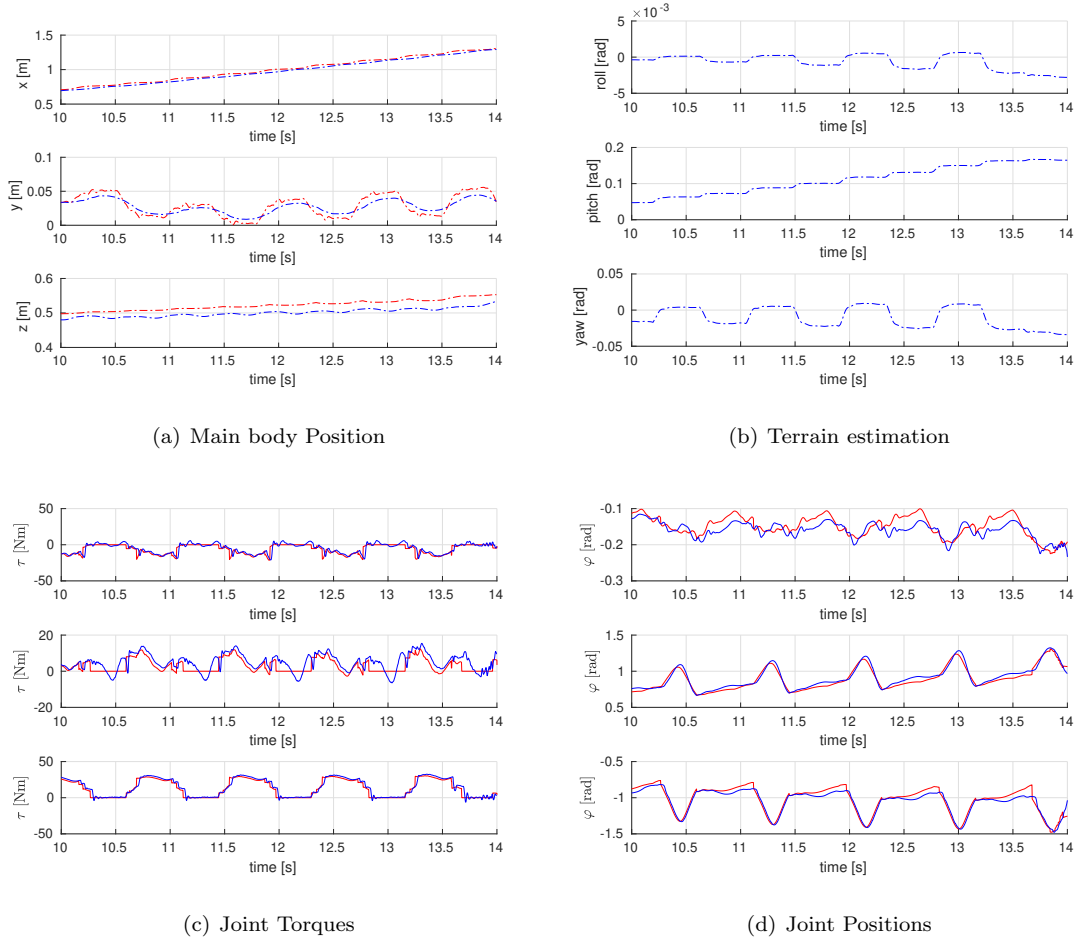(c) Joint Torques

(d) Joint Positions

Figure 4: Experimental results demonstrating ANYmal trotting up a slope. a) The tracking of the position of the torso while walking. b) The estimation of the terrain using the contact locations. c) Tracking of the joint torques for the three actuators of left-fore leg and d) the joint position tracking for the same leg. Signals depicted in red represent references quantities, while signals in blue are measured ones.

When force or contact sensors are not available, we estimate the contact forces using the equations of motion. By incorporating joint positions $\mathbf{q_j}$, velocities $\dot{\mathbf{q}}_{\mathbf{j}}$ and accelerations $\ddot{\mathbf{q}}_{\mathbf{j}}$, we can solve for $\mathbf{F_{ext}}$ by writing

$$\mathbf{F_{ext}} = (\mathbf{J_{ext}^T})^+ (\mathbf{M_j}(\mathbf{q_j})\ddot{\mathbf{q}}_{\mathbf{j}} + \mathbf{b_j}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g_j}(\mathbf{q}) - \tau) \tag{7}$$

where $\mathbf{J_{ext}} \in \mathbb{R}^{n_e \times n_{q_j}}$, $\mathbf{M_j}(\mathbf{q_j}) \in \mathbb{R}^{n_{q_j} \times n_{q_j}}$, $\mathbf{b_j}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n_{q_j}}$, $\mathbf{g_j}(\mathbf{q}) \in \mathbb{R}^{n_{q_j}}$ and τ are the geometric Jacobian of the contact location, the sub matrix of the mass matrix associated to joint accelerations only, the lower rows of the vector of Coriolis and centrifugal terms, the lower rows of the vector of gravitational terms and the joint torques, respectively. Finally, the binary contact state is obtained by thresholding $\mathbf{F_{ext}}$.

Both approaches are essential when blind locomotion is executed, but both come with their trade-offs. Using force sensors provides a direct forces measurement which can be thresholded to obtain a contact state. This measurement, however, exhibits both bias and drift which must be compensated to obtain a reliable contact state. On the other hand, using the equations of motion provides a drift-less force estimation. When faster motions are executed, a higher threshold must be employed with a method to increase the robustness in the contact estimation.

3.2 Map-Based Locomotion Planning

Locomotion planning over unknown terrain requires the integration of several modules that allow the robot to reason about its environment. The issue is even more crucial in rough terrain, where the robot might not be able to fulfill its task if the terrain properties are not taken into account. The following section shows how we combine terrain mapping and map processing, motion planning and motion control to a fully integrated system (Fankhauser, 2018).

3.2.1 Terrain Mapping and Map Processing

In our previous work (Fankhauser et al., 2018b), we have presented an approach to generate a two-and-a-half-dimensional (2.5D) elevation map by combining the robot's pose estimation and the measurements from a range sensor. This representation generates grid cells, which include height and uncertainty of the terrain. Fig. 3 (right) shows the elevation map generated by the robot while climbing a set of stairs using the robot's onboard sensor, i.e., assisted stereo camera (RealSense ZR300).

Given the continuously updated robot-centric elevation map, the data is further processed to generate more knowledge about the terrain. A surface normal vector is generated for each cell of the elevation map by using the Principal Component Analysis in a circular area around the given grid cell. To evaluate the terrain upfront the robot for feasible footholds, each cell of the elevation map is associated with a foothold score between zero and one. A linear combination of different quality measures gives an indication of whether a foothold is safe or not. The quality measures are based on geometric features like slope, curvature, roughness, and uncertainty. A three-dimension signed distance field is generated from the elevation map to plan collision-free trajectories. The signed distance field holds a distance to the nearest surface (Fankhauser et al., 2018a).

3.2.2 Motion Planning

Based on the map features described in Section 3.2.1 using the robot-centric elevation map, the robot is planning its whole-body motion according to the given terrain information. First, nominal footsteps are generated based on a geometrical approach which is recomputed after every footstep. The nominal footstep generation is based on default step lengths in longitudinal, lateral and yaw direction. By continuously recomputing the footsteps after each control loop, the footstep planner is robust against deviations from the original plan. In a next step the nominal footsteps are optimized based on the foothold scores described in Section 3.2.1. Given the nominal foothold position, the location around the foothold is checked with increasing Euclidian distance from it. If the nominal foothold position is not feasible, this approach results in a new foothold position around the nominal foothold with a valid score. Additionally, a pose optimizer verifies if the new footholds are reachable. The pose optimization takes into account a reachability cost based on kinematic limits and a cost for the static stability margin (distance between the planar projection of the Center of Mass (COM) of the support polygon). This results in a non-linear optimization problem that is solved as a Sequential Quadratic Problem (SQP). Finally, the swing leg trajectories are planned by taking the current and final foothold positions into account, as well as the signed distance field created in Section 3.2.1. For this purpose, we optimize over the knot points between the starting and end positions of the spline trajectories. This optimization results in a weighted sum of trajectory length and collision costs with the environment (Fankhauser et al., 2018a).

Given the desired trajectories of the torso and feet of the motion planner, the motion execution tries to follow these trajectories and be robust to external disturbances. The motion execution is based on the software framework Free Gait (Fankhauser et al., 2016), which internally uses the virtual model controller and contact force distribution introduced in Section 3.1.3. In this case, the controller is not using the estimated terrain from Section 3.1.1 but instead uses the terrain information from the distance sensor in Section 3.2.1. Moreover, the quadratic program that maps the net wrench acting on the torso into contact

forces is constrained by tilted friction pyramids. The normals of the friction pyramids are aligned with the surface normal vectors of the terrain (Section 3.2.1), and by taking the surface normals inside the force distribution into account the controller prevents slipping on inclined terrain (Fankhauser et al., 2018a).

4 Multi-Sensor Localization

For many applications, mission goals or waypoints are often specified as absolute coordinates in some reference frame (e.g., GPS coordinates) necessitating the robot to localize with respect to this frame. Not all localization techniques work in every environment and situation. Localizing with point cloud Iterative Closest Point (ICP) matching, for example, does not work in environments with little or highly repetitive geometric structure. For this reason, we chose to pursue a multi-sensor localization approach.

4.1 Sensors for Localization

ANYmal in its configuration used for the ERL search and rescue challenge is equipped with an RTK GPS receiver, as well as a 2D LiDAR mounted on a rotating servo-motor to obtain full 3D point clouds. These two techniques have complimentary failure modes and are therefore well-suited to achieve robust localization in different and changing environments. GPS fails when communication to satellites is degraded which typically happens inside buildings or close to high vertical structures where ICP based localization excels.

4.1.1 RTK GPS

GPS localization provides a globally consistent position and is per design drift-free. Using the RTK GPS technique its position measurement can reach centimeter-level accuracy relative to a stationary reference beacon which sends correction signals. However, since only a position measurement can be obtained, orientation drift of the proprioceptive odometry pose estimate cannot be corrected. Fig. 5 depicts the path taken by ANYmal during the ERL Challenge. The data was acquired using the onboard GPS receiver.

4.1.2 LiDAR

For localization using 3D point clouds acquired by a rotating line LiDAR sensor (Hokuyo UTM-30LX-EW), incoming point clouds are aligned to an existing map using the ICP algorithm (Pomerleau, 2013). By continuously scanning the environment, the robot is able to track its position and orientation. The individual scans (half rotation of the laser, 20'000 points) are dewarped using the local state estimation from Inertial Measurement Unit (IMU) and leg kinematics (Bloesch et al., 2012). The ICP algorithm then searches for neighbor points between the 3D point cloud of the single scan and the map (Fig. 6(a)) and tries to minimize the sum of all their distances, which takes about 0.4 s. The estimated location is then fed back to the systems state estimator as an update measurement. The challenging part of this setup is that the duration of the ICP matching step is variable. To compensate for the potentially old position update, the pose was further propagated using the local state estimation (Bloesch et al., 2012). In scenarios where an accurate point cloud map of the environment is available, this provides a drift-free 6-Degrees of Freedom (DOF) pose estimate for the robot, given that an initial alignment estimate is available (e.g., through odometry). Localization accuracy is typically centimeter-level or subcentimeter-level for position, and better than 1° for orientation.

In applications where a prior map of the environment cannot be obtained, the localization map is built incrementally by adding incoming point clouds to the map after aligning them through ICP. In these use-cases, the ICP estimate will exhibit drift according to the accumulated alignment error of incoming point clouds added to the map. The observed drift rate is mostly dependent on the incoming point clouds' overlap with the existing map, which is influenced by the LiDAR range and viewing distance.

Figure 5: An overlay of the path taken by ANYmal at the ERL Challenge with the map data provided by Google Maps. The path involved walking over sand, slopes and rough terrain.
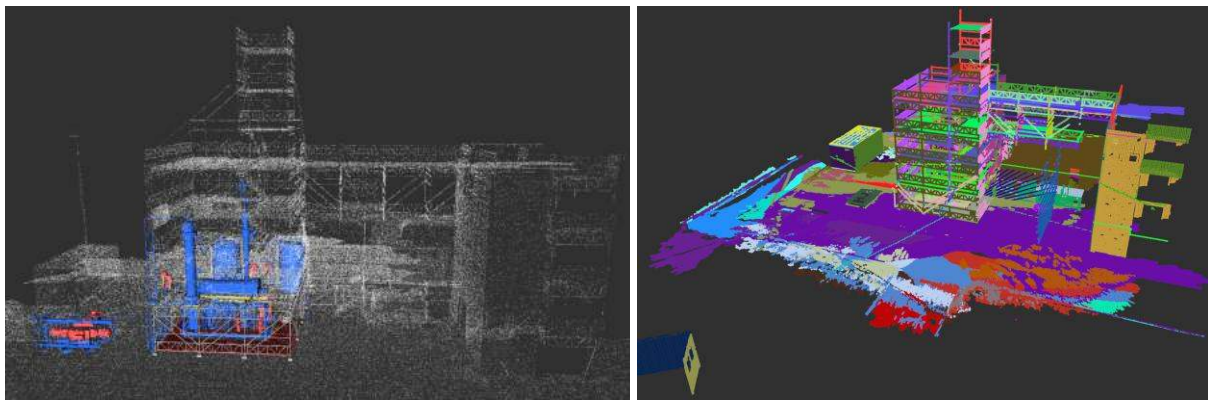
(a) Point Cloud

(b) Plane Segmentation

Figure 6: a) The ICP algorithm matches every scan acquired by the LiDAR sensor during one half rotation with the known reference map by minimizing the distance between both point clouds. b) The point cloud of the site is segmented into planes for global localization.

To converge to the correct solution, ICP requires that the errors of the initial guess of the robots pose be less than $1.5\,\text{m}$ in position and $30\,^\circ$ in orientation. In case of larger errors (i.e., at initialization or after loss of localization), the robot uses a plane matching algorithm. This algorithm searches for planes in the two 3D point clouds and groups them to all possible triples (see Fig. 6(b)). A similarity analysis of a scan and a reference plane triple can be done quickly by comparing the interplanar angles. If the plane triples are similar, their relative transformation can be directly computed from the plane parameters. Using this transformation, the scanned point cloud is expressed in the reference frame and a plausibility analysis is done by computing the nearest neighbor ratio. If a certain threshold is met, the associated relative transformation between the clouds is taken to derive the robots pose. This approach has proven to work very reliably during all missions and typically took between 20 and $100\,\text{s}$.

One caveat of ICP-based localization is that some structure in the map is required, such that the incoming point cloud can be aligned to it. This is especially problematic in large open areas with little vertical structure where alignment in horizontal direction might fail.

4.2 Sensor Fusion

Sensor fusion is needed because pose and position measurements acquired from the sources mentioned above have varying accuracy and are measured in different coordinate systems. For robot control, a smooth state estimate without jumps is desired to prevent erratic behavior of the robot. Because GPS and ICP provide absolute measurements, fusing these in a tightly-coupled fashion with the leg-odometry estimator is not an option. Therefore, we derive a loosely coupled estimation framework whose only task is to relate coordinate sensor frames to the odometry frame and to correct the drift accumulated by the odometry state estimator.

4.2.1 Coordinate Frames

We define a virtual inertial fixed frame I which will be used for path planning. The odometry state estimator estimates the pose of the robot base frame T_B in the odometry frame O. The transform corresponding to the estimated odometry pose is denoted as T_{OB}. Each sensor i measures the pose or position of the coordinate frame S_i, which is attached to the sensor, in a reference coordinate frame F_i. The coordinate sensor frame might be offset from the robot base by a transform T_{BS_i}. The pose or position transform measured by the sensor is denoted as T_{FS_i}. The sensor index i will be omitted in some of the following explanations to achieve a more readable notation. The relation of all transforms described above is shown in Fig. 7(a).

The purpose of the virtual inertial frame I is to decouple the drift dynamics of the odometry estimator from the sensor reference frames F_i. If we estimated the transform T_{OF_i} between odometry and sensor reference frame directly, a drift of the state estimator would show in all filter states which estimate these transforms, in addition to the actual dynamics of the sensor reference frame. It would also mean that we need to update all other sensor transforms if we receive a measurement from a single sensor because we gain information on the odometry drift which is encoded in all transforms T_{OF_i}.

By decoupling all frames through the intermediate inertial frame I we only need to update the transforms T_{IF_i} and T_{IO} when we receive a measurement from sensor i. Additionally, if one sensor N drops out for a period of time, we can still keep the transform T_{OF_N} up-to-date for some time, because it is decomposed into T_{IF_N} and T_{IO}. T_{IF_N} is quasi-static and T_{IO} can be estimated through updates from other sensors.

4.2.2 Implementation

We implemented the coordinate frame estimation using a Two-State Information Filter (TSIF) (Bloesch et al., 2018) which offers great model flexibility because it does not require an explicit process-model. The state vector consists of the translational and rotational part of the odometry drift transform T_{IO}, the ICP related transforms $T_{BS_{ICP}}$ and $T_{IF_{ICP}}$ and GPS transforms $T_{IF_{GPS}}$ and $T_{BS_{GPS}}$. For the GPS sensor,
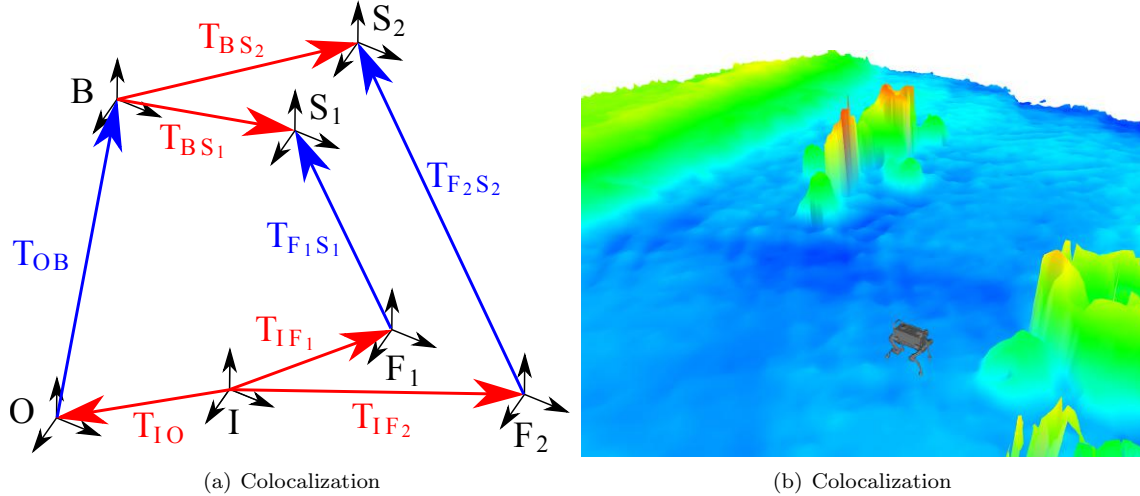
(a) Colocalization
(b) Colocalization

Figure 7: a) Coordinate frame transforms between (I)nertial, (O)dometry, (B)ase, (S)ensor-attached and sensor reference (F)rame. Transforms marked red are estimated, and blue ones are measured. b) ANYmal localized in an elevation map built by a MAV equipped with a GPS receiver.

the robot base to antenna transform $T_{BS_{GPS}}$ does not need a rotational component because the position measurement is independent of the antenna orientation. Odometry, inertial, and GPS reference frame all have their z-axis aligned with the gravity vector so that we can fix roll and pitch and only estimate the yaw angle of T_{IO} and $T_{IF_{GPS}}$. The transformation between the robot base and GPS as well as LiDAR are fixed and could be measured with sufficient accuracy. They were therefore excluded from the recursive filter update.

The TSIF is based on formulating update residuals which correlate the previous state estimate to the current state estimate. To support a multitude of different localization techniques and sensors, we defined generic residuals for position and orientation measurements, which are processed to update the coordinate transform estimates. Measurements from a pose sensor can be treated as one position and one orientation measurement. The residuals for the position update $\mathbf{\Delta}_{\mathrm{pos}}$ and the orientation update $\mathbf{\Delta}_{\mathrm{rot}}$ are defined as

$$
\begin{aligned}
\mathbf{\Delta}_{\mathrm{pos}} &= \mathbf{r}_{IO} + \mathbf{r}_{OB} + \mathbf{q}_{IO} \cdot \mathbf{q}_{OB} \cdot \mathbf{r}_{BS} - \mathbf{q}_{IF} \cdot \mathbf{r}_{FS} - \mathbf{r}_{IF} \\
\mathbf{\Delta}_{\mathrm{rot}} &= \log(\mathbf{q}_{IO} \cdot \mathbf{q}_{OB} \cdot \mathbf{q}_{BS} \cdot \mathbf{q}_{FS}^{-1} \cdot \mathbf{q}_{IF}^{-1})
\end{aligned}
\tag{8}
$$

where $\mathbf{r}_{XY}$ and $\mathbf{q}_{XY}$ are the positional and rotational component of the transform $\mathbf{T}_{XY}$ between the generic frames X and Y. The log function (Bloesch et al., 2016b) maps rotation matrices to their corresponding three-dimensional rotation vector.

5 Autonomous and Semi-Autonomous Navigation

The ability to autonomously move in challenging terrain requires that the robot can precisely (and globally) localize (Section 4), to map its environment (Section 3.2.1), to plan a navigation path, as well as to detect and overcome obstacles. The mission contains two consecutive states for *path planning* and *path following*. The *path planning* requires a start and a goal pose (whereby the start may be the robots current pose) and outputs a path. The *path following* takes this path as input and computes desired forward, sideways and rotational velocities with respect to the robots body frame for the locomotion controller described in Section 3.
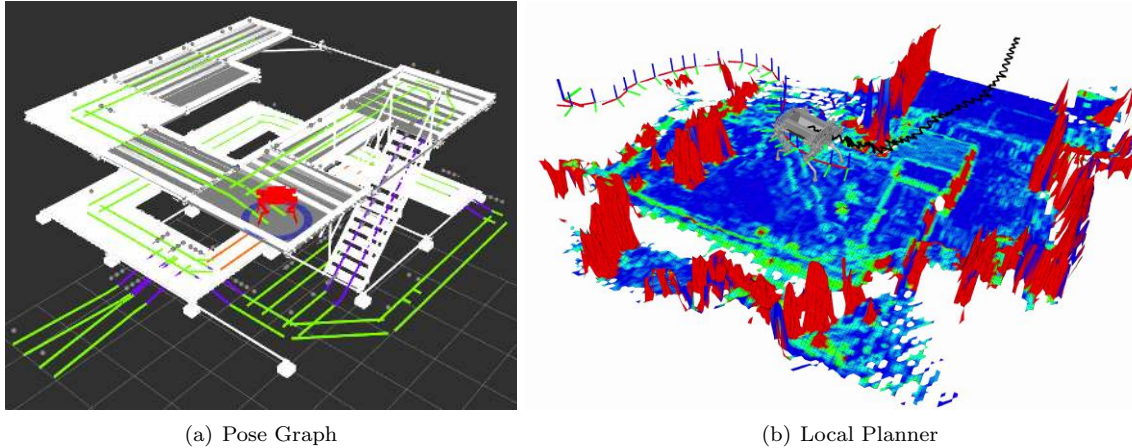
(a) Pose Graph (b) Local Planner

Figure 8: a) The pose graph consists of nodes (blue dots) and edges (green lines). b) While following the nominal trajectory, the traversability planner adapts the path around untraversable areas.

5.1 Autonomous Navigation for Industrial Inspection

Industrial inspection tasks require a fully autonomous system that is capable of detecting unforeseen obstacles. In general, the environment is known, and therefore the mission including the path can be set up in advance. Nevertheless, the robot is expected to detect local changes in the environment. For path planning, the robot features two complementary algorithms, namely the *pose graph planner* and the *traversability planner*.

The *pose graph planner* is used for global path planning, when the robot needs to navigate from one pose to another on the site. The pose graph (Fig. 8(a)), which is created once for the entire site, is a representation of the accessible and safely traversable areas on the reference map. It consists of a tree of nodes with according connections (edges). Since the individual areas are mostly flat, the node is a degenerated pose containing position and yaw information. The type of motion that the robot can execute, i.e., the type of gait or climbing maneuver, is encoded in the edge. The *pose graph planner* computes a path by using an A* algorithm on the node tree. This method has several advantages: The planning is very fast (due to the limited dimension of the problem), and the result is entirely deterministic. Any two points on the site can be connected by the pose graph planner. If a point does not lie on the pose graph, the entry or exit nodes are determined by closest proximity evaluation, even if they do not lie on it. Thanks to the short planning time, the robot can continuously and in real-time re-plan its path independent on the event of blocked paths, changed missions, or emergency situations.

For local path planning, e.g. when an obstacle blocks the global path of the robot, the *traversability planner* (Wermelinger et al., 2016)[4] comes into play. It builds up a map containing the estimated traversability of the environment, which is computed by fusing various filters for slope, roughness, or step heights using the acquired LiDAR data (Fig. 8(b)). After planning the local path with a sampling-based RRT* planner, the mission replaces the blocked segment of the global path with the alternative local path. Additionally, the robot can overcome obstacles like steps and stairs by executing the map-based locomotion planner described in Section 3.2.
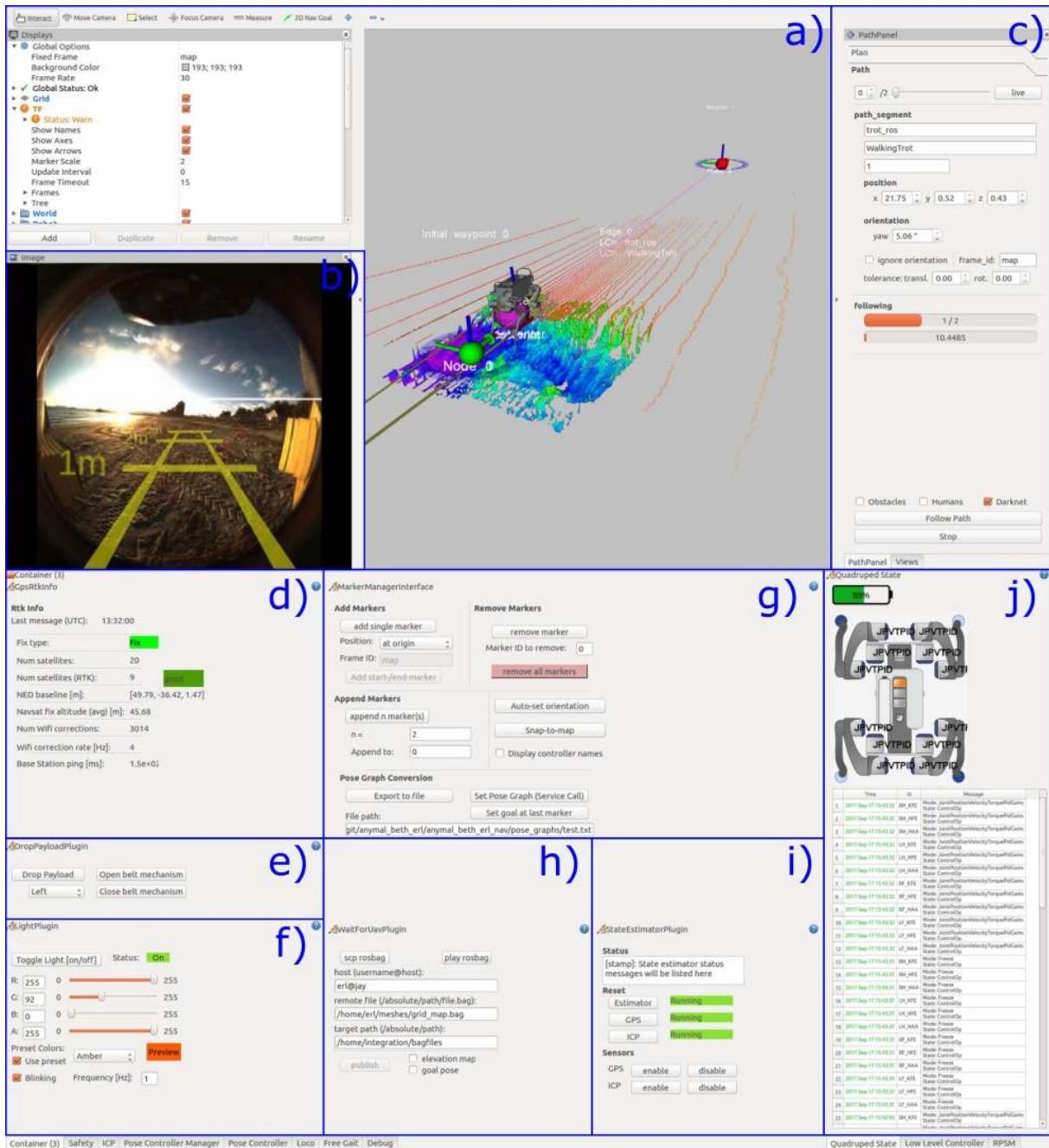
Figure 9: The GUI used throughout the ERL challenge shows the robot's status and allows the operator to interactively operate the robot through the rough environment. Thereby, a) shows the robot, the raw point cloud from the sensor unit, the elevation map, and the robot's global path. The operator can interact with the robot by adapting the global path in the three-dimensional space; b) shows the front camera's image of the robot. In addition, the yellow lines indicate the transformed robot's width and distances projected into the camera image; c) gives the operator a status of the path follower as well as the option to stop and start the path following; d) depicts an interface of the GPS RTK status; e) is an interface to drop the payload to the left or right side, respectively; f) is a tool to interact with the robot's status light; g) is the path creator tool interface the operator uses to plan and modify global paths online; h) is an interface for data exchange between multiple robots, e.g., flying and walking robot; i) allows enabling/disabling of GPS and/or ICP measurements to be considered by the frame estimation software as well as it's status and j) gives the operator a visual feedback of the actuator, battery, state estimator and contact states.

5.2   Semi-Autonomous Navigation for Search and Rescue Missions

Throughout inspection tasks, a finite set of events may be assumed, and therefore a robot may interact with such an environment in complete autonomy. In a search and rescue scenario, where several unforeseen events may happen and even be triggered by interaction with the environment, complete autonomy of an assisting robot is not yet feasible. Therefore, it is crucial to have an easy to handle and intuitive teleoperation solution, such that an operator may use the assisting robot to its full advantages. To facilitate this, a number of tools and an improved GUI (see Fig. 9) were introduced throughout the ERL challenge.

The various elements of the GUI are implemented to simplify access to different functionalities and options of the robot's software, such as streaming helper lines into a video feed, similar to the ones known in a car's backward facing parking camera. Apart from these auxiliary tools, the main improvement to the existing teleoperation framework is the introduction of a path creator tool. As described in Section 5.1 ANYmal uses a tree of graphs for efficient global planning. However, it is impractical to prepare such a graph in a search and rescue environment. Not all paths are known a priori, and new paths may appear at a later time due to the dynamic nature of such an environment. This path creator tool aims to facilitate intuitive and straightforward planning of a global path. An operator may do this by placement and subsequent interaction of markers into the robot's environmental representation. The interaction is handled via the tool's interface, a ROS compatible RQT[5] plugin. Thereby, each marker corresponds to a *node* and the path is converted online into a *pose graph* and is then used as a new global plan, on which the robot navigates.

While this tool was useful for various applications, it proved to be especially convenient in an exploration setting. While the robot is building up its map using LiDAR measurements, a human operator can interpret the camera's live video stream in parallel to the map being constructed. Since the camera looks much further ahead than the LiDAR, the operator can plan further into unknown terrain than the robot would have been able with only LiDAR data.

An additional feature of the graph creator tool is its ability to export and store planned paths and subsequently load them again. While performing different tasks, the robot may need to walk similar paths multiple times. This feature lets the operator plan a global path once, then reload and modify it as necessary for all subsequent tasks and thereby quickening the planning process substantially.

6   Collaboration with Flying Robot

An MAV is a compact and agile platform that can fly either in indoor or outdoor environments while gathering data about the surrounding area. A collaboration between the Robotic Systems Lab (RSL) and the Autonomous Systems Lab (ASL) of ETH Zurich was established to work towards the ERL challenge. The MAV focused on fusing vision-based odometry with RTK GPS to increase accuracy and robustness, and on creating a map of the environment and of the terrain from stereo vision. ANYmal could then use the terrain reconstruction to plan a traversable path and reach the desired goal: a collapsed building to inspect and explore. In order to coordinate between the robots, the same RTK system (Piksi Multi) was employed on the flying and ground robot, such that they could localize against a common coordinate frame while outdoors, and transfer maps created with the MAV to ANYmal during operation.

7   Real-World Applications for Legged Robots

Our goal is to provide a mobile platform that is versatile enough to cope with many real-world tasks. During the ERL and ARGOS challenges, we proved that the legged robot ANYmal could perform industrial

---

[4]Traversability estimation online available: https://github.com/ethz-asl/traversability_estimation
[5]Refer to http://wiki.ros.org/rqt for further information.

inspection[6] and to be used as a payload delivery machine for search and rescue[7]. While most of the robotic solutions are specialized for one specific approach, we show that ANYmal is capable of performing different missions.

7.1 Industrial Inspection

The robot can perform visual, thermal, and acoustic inspection with the payload shown in Fig. 2. While a detailed description of all tools would go beyond the scope of this paper, we want to outline the specific approach and results for pressure gauge inspection at the ARGOS challenge.

At the competition, inspection targets were marked as checkpoints in the mission description. In order to robustly point the camera at a checkpoint, the robot moves to a precomputed optimal posture at the inspection point and then determines the required pan-tilt angles online depending on the position of the robot and the checkpoint, then the algorithm switches to a tracking mode. To track the checkpoint, we use a particle filter approach. The measurement updates for the particle filter come from analyzing the image with a Histogram Oriented Gradients (HOG) (Dalal and Triggs, 2005) descriptor. For a robust detector, a global HOG descriptor is trained with machine learning, namely the SVMlight library, using about 5000 positive and negative samples. This approach has proven to work very robustly since it does not matter if the pressure gauge has a different dial face. Once the checkpoint is tracked in the image, the camera zooms in until the checkpoint has the optimal size to read.

Since it is not always possible to directly face the camera in front of the checkpoint, the image must be dewarped. Two complementary approaches have been implemented. First, a given front image of the checkpoint is matched with a given example image by looking for Scale-Invariant Feature Transform (SIFT) features (Lowe, 1999) that appear in both images. From the matched SIFT features, the homography matrix is computed and used to dewarp the image. This approach may fail when the image of the checkpoint and the example image differ too much such that there are not enough features for image matching. In this case, the algorithm can "manually" dewarp the checkpoint image knowing both the camera orientation and the checkpoint nominal orientation.

For an accurate reading of the pressure gauge, it is important to identify the center of the pressure gauge. A Hough-based circle detection is first applied to estimate the gauge frame. In a second step, a Hough-based line detection is used to find the pointer of the pressure gauge. The algorithm uses the center to filter the lines and to discard the lines that do not belong to the pointer. The mean of the resulting lines gives the estimated pointer. Finally, the actual pressure value is computed from the known scale of the pressure gauge, the center, and the line angle.

7.2 Payload Delivery for Search and Rescue

The ERL competition required the search for victims in a simulated disaster scenario and the delivery of a medkit. To complete the task, a payload delivery system was designed to securely carry the medkit (Fig. 10).

The delivery system is made of a carbon fiber plate which is rigidly attached to the upper part of the main body of ANYmal. The servo installed in the belt mechanism described in Section 2.2 can be remotely triggered via software.

This design allows implementing a simple motion procedure to deploy the medkit once the robot arrives at a target location. As illustrated in Fig. 11, ANYmal can bend its knees to lower the main body and simultaneously roll to either the left or right side. After the mechanism is triggered, the robot straightens itself back up and the procedure is terminated. The Free Gait (Fankhauser et al., 2016) Application Program

---

[6]https://youtu.be/2RQDp0Q2vSo
[7]https://youtu.be/qrJlMze_xhQ

Medkit

Belt mechanism

Figure 10: The payload delivery system, consisting of a carbon fiber plate, a belt mechanism, and a servo motor. A medical kit can be secured to the robot using the belt mechanism. This can be remotely unlatched by triggering the servo motor installed beneath the system.

Interface (API) was used to script the package drop behavior.

This mechanism worked reliably throughout all tests and is well suited for single packet delivery. However, if multiple objects need to be delivered, a different mechanism should be applied.

8 Conclusions

The present paper illustrates the worldwide first attempt at using a versatile legged machine for autonomous inspection on industrial sites. It also outlines our approach in using such a machine for a realistic search and rescue scenario.

In three consecutive ARGOS competitions on a testing site in Pau, France, our team was able to demonstrate the high potential of the proposed solution. While all individual tasks regarding system mobility, navigation, and inspection could be entirely fulfilled, the high system complexity entails several potential sources for failure. Firstly, the fact that neither stopping (blocking) nor disabling the joint motors leads to an immediate stop of a legged robot complicates safety considerations. As a result, we encountered several (uncontrolled) falls during the three competitions, all of which the robot survived. In the future, it is required to further extend the work on smart emergency behaviors such that legged robots can safely operate even in case of critical software or hardware failure. Secondly, the realistic missions unveiled that robust, reliable, and fast terrain perception under harsh conditions is still challenging. When the robot was moving fast, it was difficult to create terrain maps without any false positive obstacles due to the distorted scans from the motion, reflections on metal and wet surfaces, and water drops on the laser. Thirdly, despite the superiority in mobility compared to tracked or wheeled vehicles, there is still a lot of potential for improvement of the locomotion skills of legged robots. Once these deficiencies are overcome, we are convinced that legged robots such as ANYmal can find their way into applications like industrial inspection.
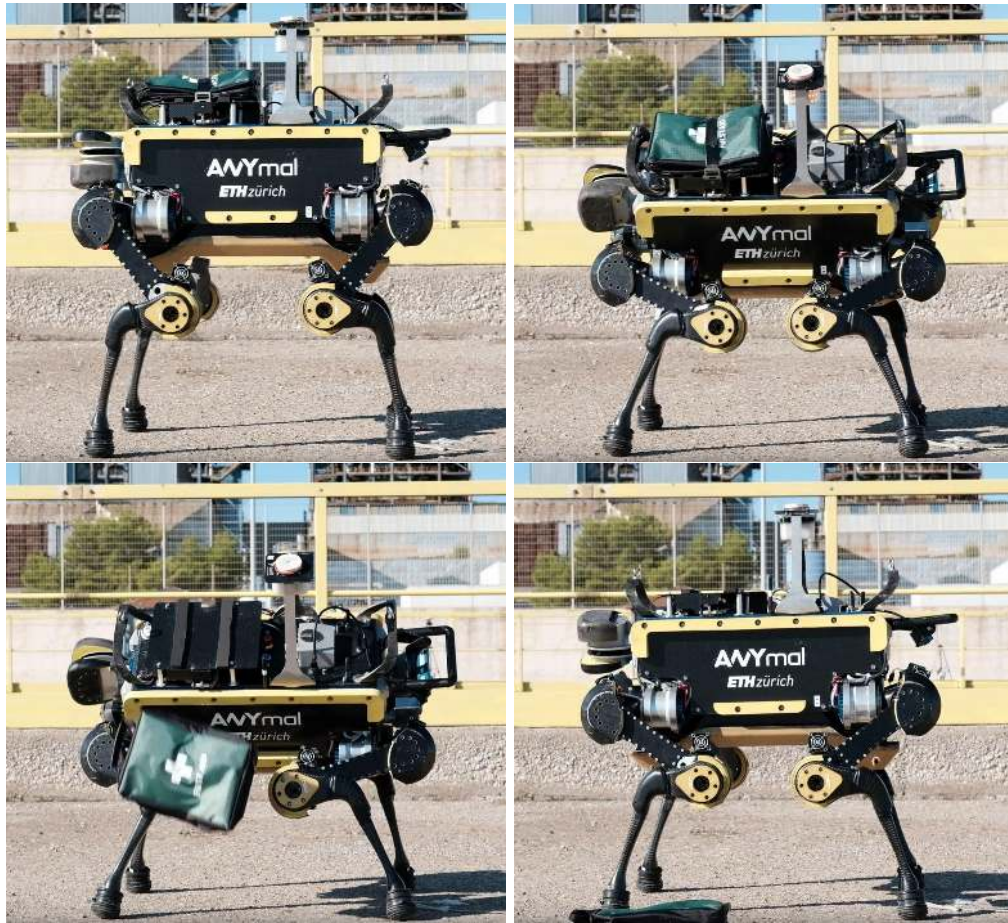
Figure 11: The payload delivery procedure. From its initial state (top left) the robot starts to tilt sideways (top right) until a certain roll angle is achieved. Afterward, the payload delivery mechanism releases its belt strip to drop the payload (lower left), and the robot straightens back up (lower right).

Similarly, during the ERL challenge, team ETH Zurich demonstrated the potential of the use of a legged robotic system for payload delivery. Using a GPS RTK receiver to globally localize itself, the robot was able to follow high-level target poses to navigate towards the competition goal in a semi-autonomous fashion. The advances in our control framework enabled ANYmal to locomote robustly in very rough terrain, including walking in shallow seawater, over wet and dry sand, sticks and steep slopes. A novel localization framework was implemented to fuse together different pose estimation sources, i.e., GPS measurements and pose estimates originating from LiDAR scan matching. A custom payload delivery system was developed which could be remotely triggered to carry and drop a medkit once a target pose was reached.

Acknowledgments

References

Bloesch, M., Burri, M., Sommer, H., Siegwart, R., and Hutter, M. (2018). The Two-State Implicit Filter Recursive Estimation for Mobile Robots. *IEEE Robotics and Automation Letters*, (accepted).

Bloesch, M., Hutter, M., Hoepflinger, M., Leutenegger, S., Gehring, C., Remy, C. D., and Siegwart, R. (2012). State Estimation for Legged Robots - Consistent Fusion of Leg Kinematics and IMU. In *Robotics Science and Systems (RSS)*, pages 17–24.

Bloesch, M., Sommer, H., Laidlow, T., Burri, M., Nuetzi, G., Fankhauser, P., Bellicoso, D., Gehring, C., Leutenegger, S., Hutter, M., et al. (2016a). A primer on the differential calculus of 3d orientations. *arXiv preprint arXiv:1606.05285*.

Bloesch, M., Sommer, H., Laidlow, T., Burri, M., Nuetzi, G., Fankhauser, P., Bellicoso, D., Gehring, C., Leutenegger, S., Hutter, M., and Siegwart, R. (2016b). A Primer on the Differential Calculus of 3D Orientations. Technical report.

Dalal, N. and Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. In *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 886–893.

Davids, A. (2002). Urban search and rescue robots: from tragedy to technology. *IEEE Intelligent systems*, 17(2):81–83.

Fankhauser, P. (2018). *Perceptive Locomotion for Legged Robots in Rough Terrain*. Doctoral thesis, ETH Zurich.

Fankhauser, P., Bellicoso, C. D., Gehring, C., Dube, R., Gawel, A., and Hutter, M. (2016). Free Gait An architecture for the versatile control of legged robots. In *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 1052–1058. IEEE.

Fankhauser, P., Bjelonic, M., Takahiro, M., Baumann, T., Bellicoso, C. D., Gehring, C., and Hutter, M. (2018a). Robust rough-terrain locomotion with a quadrupedal robot. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Fankhauser, P., Bloesch, M., and Hutter, M. (2018b). Probabilistic terrain mapping for mobile robots with uncertain localization. PP:1–1.

Feng, S., Whitman, E., Xinjilefu, X., and Atkeson, C. G. (2015). Optimization-based full body control for the darpa robotics challenge. *Journal of Field Robotics*, 32(2):293–312.

Gehring, C., Bellicoso, C. D., Coros, S., Bloesch, M., Fankhauser, P., Hutter, M., and Siegwart, R. (2015). Dynamic trotting on slopes for quadrupedal robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5129–5135. IEEE.

Gehring, C., Coros, S., Hutler, M., Bellicoso, C. D., Heijnen, H., Diethelm, R., Bloesch, M., Fankhauser, P., Hwangbo, J., Hoepflinger, M., et al. (2016). Practice makes perfect: An optimization-based approach to controlling agile motions for a quadruped robot. *IEEE Robotics & Automation Magazine*, 23(1):34–43.

Hutter, M., Bodie, K., Lauber, A., and Hwangbo, J. (2016a). EP16181251 - Joint unit, joint system, robot for manipulation and/or transportation, robotic exoskeleton system and method for manipulation and/or transportation. European Patent Application 16181251.6, 26.07.2016.

Hutter, M., Diethelm, R., Bachmann, S., Fankhauser, P., Gehring, C., Tsounis, V., Lauber, A., Guenther, F., Bjelonic, M., Isler, L., et al. (2018). Towards a generic solution for inspection of industrial sites. In *Field and Service Robotics*, pages 575–589. Springer.

Hutter, M., Gehring, C., Bloesch, M., Hoepflinger, M. A., Remy, C. D., and Siegwart, R. (2012). Starleth: A compliant quadrupedal robot for fast, efficient, and versatile locomotion. In *Adaptive Mobile Robotics*. World Scientific.

Hutter, M., Gehring, C., Jud, D., Lauber, A., Bellicoso, C. D., Tsounis, V., Hwangbo, J., Bodie, K., Fankhauser, P., Bloesch, M., Diethelm, R., Bachmann, S., Melzer, A., and Hoepflinger, M. (2016b). ANYmal - a highly mobile and dynamic quadrupedal robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 38–44. IEEE.

Johnson, M., Shrewsbury, B., Bertrand, S., Wu, T., Duran, D., Floyd, M., Abeles, P., Stephen, D., Mertins, N., Lesman, A., et al. (2015). Team ihmc's lessons learned from the darpa robotics challenge trials. *Journal of Field Robotics*, 32(2):192–208.

JPT Staff (2012). Sensabot: A Safe and Cost-Effective Inspection Solution. *Journal of Petroleum Technology*, 64(10):32–34.

Kaneko, K., Morisawa, M., Kajita, S., Nakaoka, S., Sakaguchi, T., Cisneros, R., and Kanehiro, F. (2015). Humanoid robot hrp-2kaiimprovement of hrp-2 towards disaster response tasks. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pages 132–139. IEEE.

Keselman, L., Woodfill, J. I., Grunnet-Jepsen, A., and Bhowmik, A. (2017). Intel realsense stereoscopic depth cameras. *arXiv preprint arXiv:1705.05548*.

Kolvenbach, H. and Hutter, M. (2017). Life Extension: An Autonomous Docking Station for Recharging Quadrupedal Robots. In *(submitted to) Field and Service Robots (FSR)*.

Kuindersma, S., Deits, R., Fallon, M., Valenzuela, A., Dai, H., Permenter, F., Koolen, T., Marion, P., and Tedrake, R. (2016). Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Autonomous Robots*, 40(3):429–455.

Kydd, K., Macrez, S., and Pourcel, P. (2015). Autonomous Robot for Gas and Oil Sites. In *SPE Offshore Europe Conference and Exhibition*. Society of Petroleum Engineers.

Lowe, D. (1999). Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision*, pages 1150–1157. IEEE.

Murphy, R. R., Tadokoro, S., Nardi, D., Jacoff, A., Fiorini, P., Choset, H., and Erkmen, A. M. (2008). Search and rescue robotics. In *Springer Handbook of Robotics*, pages 1151–1173. Springer.

Park, H.-W., Wensing, P. M., and Kim, S. (2017). High-speed bounding with the mit cheetah 2: Control design and experiments. *The International Journal of Robotics Research*, 36(2):167–192.

Pfeiffer, K., Bengel, M., and Bubeck, A. (2011). Offshore robotics - Survey, implementation, outlook. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 241–246.

Pomerleau, F. (2013). *Applied registration for robotics*. PhD thesis, ETH.

Pratt, G. and Manzo, J. (2013). The darpa robotics challenge [competitions]. *IEEE Robotics & Automation Magazine*, 20(2):10–12.

Raibert, M., Blankespoor, K., Nelson, G., and Playter, R. (2008). Bigdog, the rough-terrain quadruped robot. *IFAC Proceedings Volumes*.

Röning, J., Kauppinen, M., Pitkänen, V., Kemppainen, A., and Tikanmäki, A. (2017). The challenge of preparing teams for the european robotics league: Emergency. *Electronic Imaging*, 2017(9):22–30.

Semini, C., Barasuol, V., Goldsmith, J., Frigerio, M., Focchi, M., Gao, Y., and Caldwell, D. G. (2017). Design of the hydraulically actuated, torque-controlled quadruped robot hyq2max. *IEEE/ASME Transactions on Mechatronics*.

Semini, C., Tsagarakis, N. G., Guglielmino, E., Focchi, M., Cannella, F., and Caldwell, D. G. (2011). Design of HyQ – a hydraulically and electrically actuated quadruped robot. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 225(6):831–849.

Seok, S., Wang, A., Chuah, M. Y. M., Hyun, D. J., Lee, J., Otten, D. M., Lang, J. H., and Kim, S. (2015). Design principles for energy-efficient legged locomotion and implementation on the mit cheetah robot. *IEEE/ASME Transactions on Mechatronics*, 20(3):1117–1129.

Tsagarakis, N. G., Caldwell, D. G., Negrello, F., Choi, W., Baccelliere, L., Loc, V., Noorden, J., Muratore, L., Margan, A., Cardellino, A., et al. (2017). Walk-man: A high-performance humanoid platform for realistic environments. *Journal of Field Robotics*, 34(7):1225–1259.

Wermelinger, M., Fankhauser, P., Diethelm, R., Krusi, P., Siegwart, R., and Hutter, M. (2016). Navigation planning for legged robots in challenging terrain. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1184–1189. IEEE.

Winfield, A. F., Franco, M. P., Brueggemann, B., Castro, A., Limon, M. C., Ferri, G., Ferreira, F., Liu, X., Petillot, Y., Roning, J., et al. (2016). eurathlon 2015: A multi-domain multi-robot grand challenge for search and rescue robots. In *Conference Towards Autonomous Robotic Systems*, pages 351–363. Springer.

Zucker, M., Joo, S., Grey, M. X., Rasmussen, C., Huang, E., Stilman, M., and Bobick, A. (2015). A general-purpose system for teleoperation of the drc-hubo humanoid robot. *Journal of Field Robotics*, 32(3):336–351.