Intel Serv Robotics (2014) 7:67–77
DOI 10.1007/s11370-014-0147-9

SPECIAL ISSUE

Quadruped robot trotting over irregular terrain assisted by stereo-vision

Stéphane Bazeille · Victor Barasuol · Michele Focchi · Ioannis Havoutis · Marco Frigerio · Jonas Buchli · Darwin G. Caldwell · Claudio Semini

Received: 10 August 2013 / Accepted: 4 February 2014 / Published online: 7 March 2014

Abstract Legged robots have the potential to navigate in challenging terrain, and thus to exceed the mobility of wheeled vehicles. However, their control is more difficult as legged robots need to deal with foothold computation, leg trajectories and posture control in order to achieve successful navigation. In this paper, we present a new framework for the hydraulic quadruped robot HyQ, which performs goal-oriented navigation on unknown rough terrain using inertial measurement data and stereo-vision. This work uses our previously presented reactive controller framework with balancing control and extends it with visual feedback to enable closed-loop gait adjustment. On one hand, the camera images are used to keep the robot walking towards a visual target by correcting its heading angle if the robot deviates from it. On the other hand, the stereo camera is used to estimate the size of the obstacles on the ground plane and thus the terrain roughness. The locomotion controller then adjusts the step height and the velocity according to the size of the obstacles. This results in a robust and autonomous goal-oriented navigation over difficult terrain while subject to disturbances from the ground irregularities or external forces. Indoor and outdoor experiments with our quadruped robot show the effectiveness of this framework.

Keywords Reactive walking · Active impedance · Goal-oriented navigation · Visual servoing · Quadruped robot

S. Bazeille (✉) · V. Barasuol · M. Focchi · I. Havoutis · M. Frigerio · D. G. Caldwell · C. Semini
Department of Advanced Robotics, Istituto Italiano di Tecnologia (IIT), via Morego, 30, 16163 Genova, Italy
e-mail: stephane.bazeille@gmail.com

V. Barasuol
e-mail: victor.barasuol@iit.it

M. Focchi
e-mail: michele.focchi@iit.it

I. Havoutis
e-mail: ioannis.havoutis@iit.it

M. Frigerio
e-mail: marco.frigerio@iit.it

D. G. Caldwell
e-mail: darwin.caldwell@iit.it

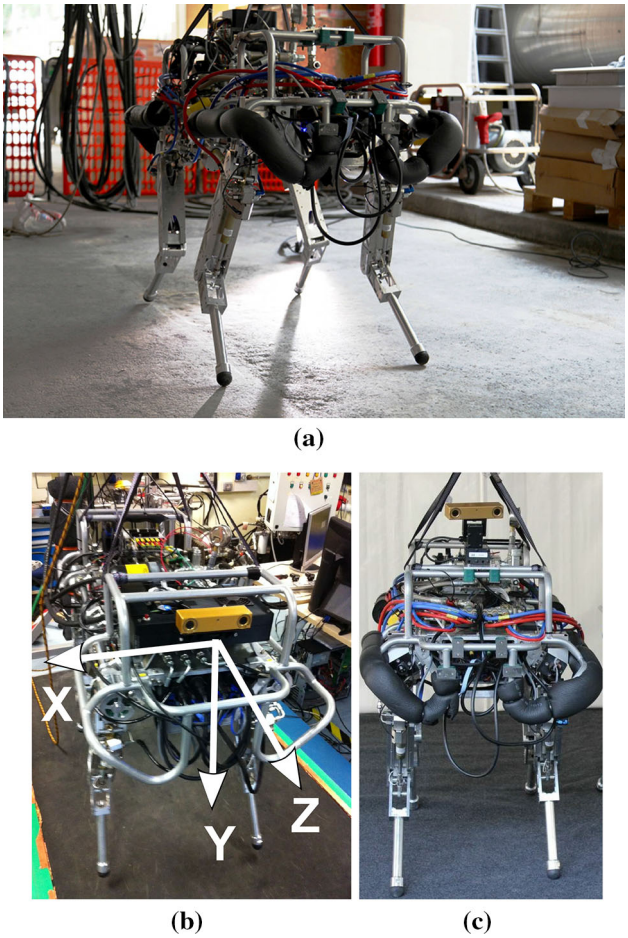
C. Semini
e-mail: claudio.semini@iit.it

J. Buchli
Agile and Dexterous Robotics Lab, ETH Zurich, Tannenstr. 3, 8092 Zurich, Switzerland
e-mail: buchlij@ethz.ch

1 Introduction

Legged locomotion is a complex task for robots, involving different components ranging from low-level motor control to high-level cognitive processes. To be autonomous, robots need all these components to be reliable, well orchestrated and capable of real-time execution. The hydraulic quadruped, HyQ (Fig. 1) is a versatile robot with hydraulic actuation developed at the Department of Advanced Robotics at the Istituto Italiano di Tecnologia (IIT) [22]. HyQ is fast, robust, fully torque controllable, actively compliant and built for dynamic locomotion.

Our previous work focused on dynamic locomotion, mainly trotting, using active impedance and low-level feedback from the on-board inertial measurement unit (IMU) for stabilization [3,12,24]. Such low-level control can reliably negotiate flat and moderately rough terrain (obstacles lower than 10 cm) while following manually selected high-level parameters, i.e., velocity, heading, and step height. This provides a solid foundation for building up a set of higher-level



(a) (b) (c)

Fig. 1 Pictures of IIT's quadruped robot HyQ. **a** Without stereo camera (2012); **b** with the stereo camera fixed on the protection frame (2012); **c** with the stereo camera mounted on a pan and tilt unit (2013). **b** shows the definition of the camera coordinate frame.

controllers that deal with the cognitive aspects of locomotion and navigation, to further increase HyQ's autonomy. Following this theme, we added a stereo-vision system as a first step towards providing the robot with higher-level feedback, which can in turn be used in a number of ways, e.g., localization, mapping and path planning.

Visual feedback is crucial in the context of autonomy in real-world scenarios where open-loop approaches, e.g., dead reckoning, quickly accumulate errors due to foot slippage, non-uniform weight distribution, terrain irregularities or external disturbances on the body.

This work is an extension of our previously presented reactive controller framework [3], now extended with the addition of visual feedback. The vision system sends to the controller a qualitative localization and information about the terrain to autonomously and continuously adapt the trotting parameters, i.e., heading, forward velocity, step height and duty factor.

This visual feedback allows to guide the robot towards a visual goal while traversing challenging terrain in the presence of external disturbances. Such disturbances can be created by lateral pushes on the robot, foot slippage or foot-object frontal impacts. Also, it allows us to compensate for possible lateral drift due to inaccurate calibration or transient loss of balance. Furthermore, it makes the behavior safer by detecting obstacles, slowing down when necessary, increasing the duty factor or the step height of the trot to overcome obstacles and in the worst case stopping the robot in front of an obstacle that cannot be avoided.

1.1 Contribution

A new reactive controller using position, force, inertial measurements and vision data for closed-loop gait adjustment. The focus of this paper lies on the controller that allows a highly dynamic quadrupedal robot to perform a fully autonomous reactive trot in an unknown irregular terrain.

1.2 Contents

The structure of the paper is organized as follows. In Sect. 2, we present a review of related work on quadruped robot navigation, and in Sect. 3, we provide details about our perception algorithms. Section 4 describes the locomotion controller with visual feedback. In Sect. 5, we present our quadruped robot and the results of indoor and outdoor experiments. Finally, Sect. 6 discusses the results and Sect. 7 concludes the paper and mentions future work.

2 Related work

Quadrupedal locomotion has been an active area of robotics research for several decades. However, up to now few people have worked on the integration of vision sensors on quadrupedal platforms. Such platforms are commonly used to develop low-level controllers, rather than high-level cognitive processes.

A number of studies in quadrupedal locomotion often simplify the problem of perception using accurate a-priori given maps and external robot state sensors. For example the standard test set up of the DARPA Learning Locomotion Program used pre-scanned maps and a marker-based tracking system on LittleDog [15,20].

Kolter et al. in [16] presented a more autonomous approach by removing the dependence on given maps and external state input. In their control framework they use a stereo camera together with a well-established point-cloud matching technique to iteratively build a map of the environment that is then used for navigation. While the camera was on the robot, the vision processing and path planning were calculated on an external computer.

Filitchkin and Byl used a monocular camera to perform terrain classification that in turn influences the locomotion behavior of their LittleDog quadruped [8].

Chilian and Hirschmuller in [7] performed position estimation and terrain modeling using a stereo camera on their hexapod robot, while Shao et al. [25] presented an obstacle avoidance approach for their quadruped robot that uses a stereo-vision-based terrain modeling algorithm.

Howard in [13] introduced a state estimation approach for BigDog that combines a number of different sensor modalities, including stereo camera, IMU, odometry and GPS to achieve accurate long-term positioning.

Bajracharya et al. recently showed terrain mapping for vision-in-the-loop walking on the LS3 robot from Boston Dynamics [2]. The main contribution is the robustness of the mapping in difficult terrain (vegetation, slopes) and difficult lightning condition (day or night). The vision system is used to map the environment in the vicinity of the robot and inform the gait generation process about possible changes in the surface where the robot is locomoting. As with other legged robots of Boston Dynamics, very few details on the controller, hardware, and experimental data have been scientifically published so far. Therefore, the results of the experiments shown in online videos are notoriously hard to scientifically verify and compare.

Research in a similar direction was also performed on the AIBO quadruped entertainment robot designed and manufactured by Sony. AIBO was the first quadruped robot with an on-board camera that was able to detect a number of objects and to track a pink ball while navigating [10].

Our work differs from the literature as we propose a new reactive controller that closes the loop with vision feedback to continuously adjust gait parameters. In other words, we present an approach without mapping or path planning, where all the computation is done on-board. Typical methods for robots maintain a local model of the terrain near the robot and localize it within that estimated model. Unfortunately, those methods rely on well orchestrated, complex high-level processes such as SLAM, state estimation and path planning and require consistency and heavy computation with real-time processing. On HyQ, it is difficult to realize in terms of robustness and computing speed as the robot is subject to considerable motions (due to highly dynamic manoeuvres, impacts, vibrations, slippages) during locomotion. We therefore proposed to give to the robot only perception feedback without mapping to keep the robot's ability to react quickly. Our visual process requires small computational effort, which allows an implementation of all computation on-board. The process sends at 15 Hz a qualitative localization and information about the terrain to the reactive controller, which then performs closed-loop gait adjustment. The stereo-vision system feeds back the position of the target the robot is approaching and the distance to it. Furthermore, it extracts and transmits the height of obstacles in front of the robot and the distance to them. Thus, the visual feedback allows the robot to estimate the terrain difficulty ahead of time and therefore perform gait adjustment in advance, instead of purely reacting to terrain changes.

3 Environment perception

In our previous work we developed motion control algorithms based on joint positions/velocities and the body state information given by the IMU. The IMU was the first perception sensor we added to our quadruped platform to provide relative information between the robot and the world. However, the robot's orientation in the world frame alone is not enough to create cognitive interaction, information about the environment itself are needed. To perceive the environment and to improve the locomotion robustness we therefore added a stereo camera to the robot. The HyQ stereo camera set up uses a Bumblebee2 firewire color camera from point grey. It has a focal length of 2.5 mm, a field of view of 97 degrees, a maximum resolution of 1,024 $\times$ 768 at 20 fps, a 12 cm baseline, and it is pre-calibrated against distortions and misalignment. On our system, a point cloud with 640 $\times$ 480 3D points with their associated RGB values can be computed at 15 Hz on a dedicated vision computer equipped with a quad-core Intel processor at 2.50 GHz running Ubuntu.

Four parameters are extracted from the images and sent to the motion planner: the position of the target in the image frame, the distance to this target, and the height of and the distance to the highest obstacle in front of the robot. The first two corresponds to a qualitative localization, and the third and fourth are information about the terrain.

3.1 Target tracking

For the tracking we decided to use the color information with the Mean Shift algorithm. It was the most intuitive way as we followed a target for the heading control. It does not necessitate any learning stage or parametrization, and the target can be any objects selected manually by the operator in the image before starting the navigation. Our method has been successfully used indoors and outdoors (natural light) during short experiments but for more robustness in outdoor settings and long run navigation it has to be noted that tracking SIFT features [17] will improve robustness. The method proposed by Zhou et al. [26] also based on Mean Shift would be well suited for our purpose and would avoid any color tracking problems.

We implemented a modified version of the CAMShift algorithm [6]. Camshift (continuously adaptive mean shift) combines the basic mean shift algorithm with an adaptive region-sizing step. A review on Mean Shift methods used for

tracking can be found in [1]. In this method, the kernel is a simple step function applied to a color probability map. The color probability of each pixel is computed using a histogram back projection. The algorithm creates a confidence map in the new image based on the color histogram of the object in the previous image, and uses mean shift to find the peak of a confidence map near the object's old position.

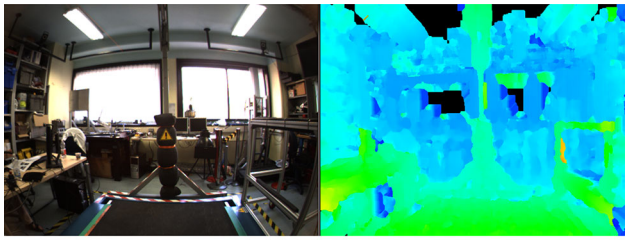
Color is represented as Hue from the HSV color model, a color space that is really more consistent than the standard RGB color space under illumination changes. Since Hue is unstable at low saturation, the color histograms do not include pixels with saturation below a threshold.We use a successively opening and closing to filter the back projection image (see Fig. 3e) to remove the outliers and to enhance the object. The post filtering gives better segmented images in case of complex video sequences (e.g., changes in illumination, appearance, scale or object movement). Moreover, as the body of the quadruped robot trunk is subject to considerable movements during locomotion, we increased the search region for the Mean Shift to make the tracking more robust under real trotting conditions.

In Fig. 3, we show the detection of a red target indoors. The tracking was achieved using the left camera with a 15Hz frame rate. The implemented tracking has been tried on different objects with colors different from the background and with a minimum size of 10 cm and under different conditions (indoors, outdoors, artificial or natural light). Practically, we were remotely selecting in the images an object present in the scene as goal for the robot. In the experiment shown in Fig. 2 we were tracking a yellow sign attached to the crane in front of the robot, and in Figs. 13 and 14 we tracked a red toolbox in outdoor conditions.

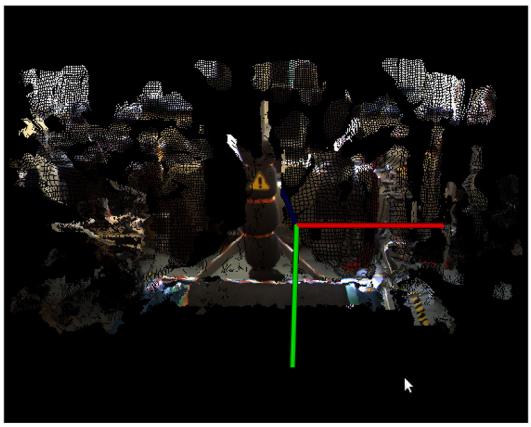
3.2 Depth map and height map from stereo images

We get images from a stereo camera to obtain two different views of the scene. By matching the images, the relative depth information can be obtained as a disparity map, which is inversely proportional to the differences in distance to the objects. The disparity map refers to the difference in x coordinates of similar features within two stereo-rectified images. The reader can refer to Hartley and Zisserman book for more details on getting depth from stereo-vision [11].

As this camera is accurately pre-calibrated against distortions and misalignment (the stereo pair are aligned within 0.11 pixel RMS error), high-quality rectified images are extracted using Point Grey proprietary library Triclops. That results in accurate correspondence computation between the two stereo images. The computation of the correspondences is achieved using the SAD method [18] on edge images. It allows the matching on the changes in brightness rather than the absolute values of the pixels in the images which is more robust in environment where the lighting conditions change.



(a)



(b)

Fig. 2 **a** *Right image* and *colored* disparity image; **b** associated point cloud (about 100,000 points) with the camera reference frame: x in *red*, y in *green* and z in *blue* (color figure online)

A good precision is obtained on the disparity map as we use sub-pixel interpolation and a size of the matching mask set to 21×21. When the disparity map is processed we use surface validation [19] to remove outliers. Also, we apply a 3×3 median filtering to fill small holes. An example of the post-processed disparity map is shown in Fig. 2a. The disparity is shown with colors (hot and cold color bar) to appreciate its quality. It has to be noted that the quality of this disparity is important for the depth image computation. Outliers and missing values are low as shown in Fig. 2a. Then for each valid disparity pixel we can estimate the corresponding 3D position.

The depth of all those point gives the depth map (Fig. 3c) and the height to which we subtract the robot height computed from the legs position gives the height map (Fig. 4).

3.3 Controller input

The vision system provides visual data with a frame rate of 15 Hz to the robot controller. The visual data packet contains the 3D position vector of the tracked object and the height of and the distance to the highest obstacle in front of the robot.

The 3D position vector of the tracked object is composed of the x and y positions of the barycenter of the ellipse in the camera frame and the distance to the target. The distance to the target is computed by extracting all the pixels of the

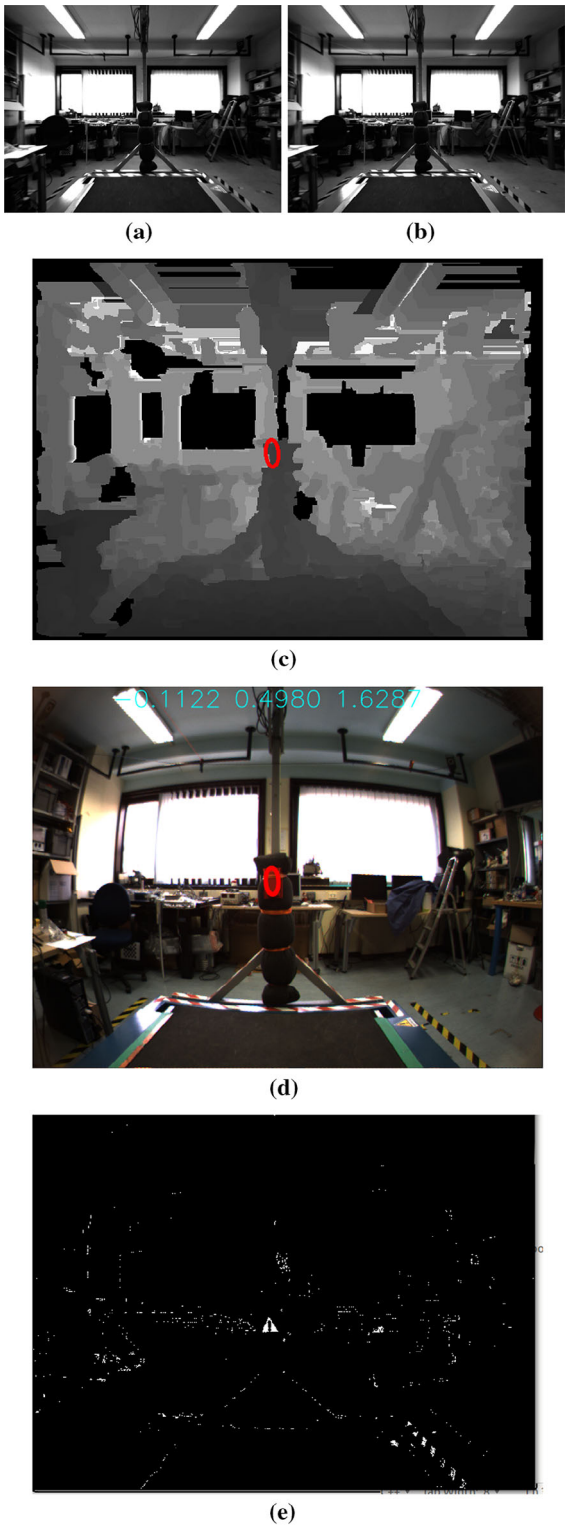


Fig. 3 **a**, **b** *Left* and *right* rectified images used to compute the depth map; **c** depth map showing the tracked *colored* object with a *red oval* in the center of the image; **d** example of color detection of a *red box*. This object is tracked continuously and its 3D position (expressed in meters in the camera frame defined in Fig. 1b) is displayed in *blue* in the *top* of the image; **d** back projection image before post-processing (color figure online)

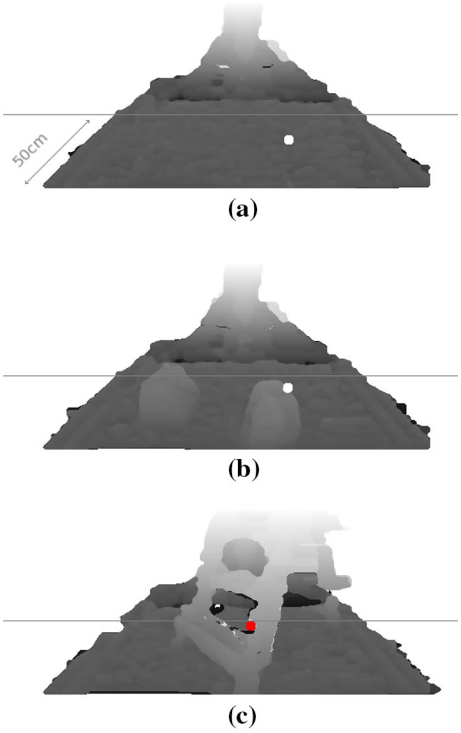


Fig. 4 Example of height map in the robot frame. *Darks colors* mean 0 height and white colors stands for the robot's height. **a** Without obstacles (corresponding to the stereo pair Fig. 3a, b); **b** with rocks and **c** with an obstacle that we cannot cross. The *white* and *red dots* in the map represent the highest points. The computed height were respectively 0.70, 0.63, 0.20 m (y axis). The *grey line* represents the 1.5 m distance limit (z axis) from the robot as shown in Fig. 5 (color figure online)

depth map which belong to the ellipse and looking for the median value.

The height of the highest obstacle and its distance to the robot are computed by identifying the highest point in a given area in the depth map. This area is defined as 2×0.5 m at 1 m in front of the robot see Fig. 5). Figure 4 shows an example of height map. The maximum height is computed as the distance between the top of the obstacle from a horizontal plane spanned by the center points of the foot trajectories (cfg. Sect. 4). The accuracy of the obstacle height is about ± 2 and ± 5 cm for the distance. In the case the obstacle can be crossed (this is determined by the maximum retraction capability of the robot leg, in our case 25 cm), the step height is modified accordingly.

All these values are expressed in the camera reference frame and are translated into the robot base frame (via an appropriate homogeneous transform) and are temporally filtered before being sent to the robot controller to smooth the robot behavior and filter small oscillations or outliers. It has to be noted that the values of the height after being filtered are processed through the variable delay digital buffer detailed in the next section.

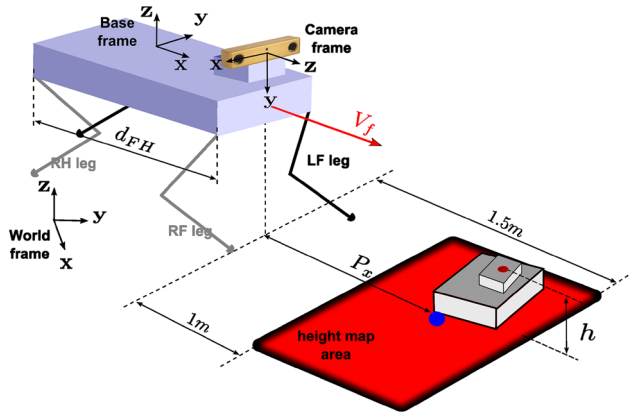


Fig. 5 Definition of HyQ reference frames. The height map area is shown in *light red color*. The *red dot* indicates the highest obstacle, while the *blue dot* represent the *center* of the area. The distance P_x from the robot and this point in the *horizontal* direction is $P_x = 1.25m$ (color figure online)

Special cases In case of tracking problem, i.e., tracking lost for example, special values are sent and the robot stop. The robustness of this computation is discussed further in Sect. 6.

In case the obstacle can be crossed because its height is higher than maximum leg lift, the robot stops in place. It is worth mentioning that this robot could possibly overcome bigger obstacles if a different locomotion strategy is considered (e.g., jumping or climbing), but we consider only a trotting gait in this study.

3.4 Obstacle buffer

When the computed maximum height of the obstacle and its distance to the robot are sent to the locomotion controller, it has to modify the leg lift accordingly to make the robot able to overcome it. However, when the robot approaches the obstacle, this goes out of the vision field of the camera and the estimation of the distance from visual information alone becomes impossible. Therefore, the robot keeps a "memory" of the height map which is stored in a buffer. In particular a delay is introduced in order to modify the step height at the moment in which the robot is approaching a certain obstacle. This delay between detection and application of the height modification is dependent on the forward velocity V_f of the robot which is estimated using the leg odometry.

A variable delay digital buffer is implemented to take this fact into account (see Fig. 6). As soon the height data ($h(t)$) is coming from the vision (for our experiments at a 15 Hz rate) it is stored in a buffer (Fig. 7). An appropriate saturation function limits the values of $h(t)$ to avoid commanding motions to the feet which are out of the workspace. The first element of the buffer is the actual sample coming from the vision with no delay. The size of the buffer N_{buf} is computed

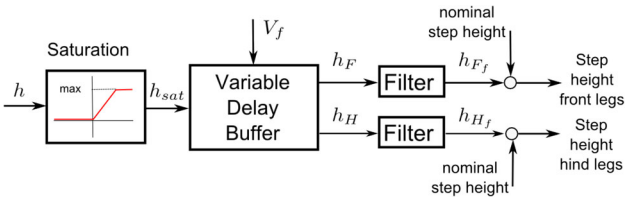


Fig. 6 Schematic of the variable delay buffer approach. The modification of the step height is delayed depending on the distance and the robot speed V_f

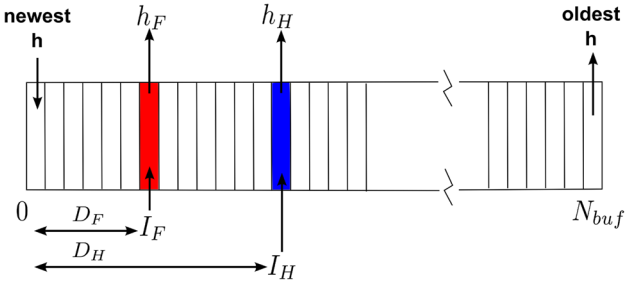


Fig. 7 Schematic of the indexing of the buffer. Indexes can move forward or backward according to the robot speed V_f

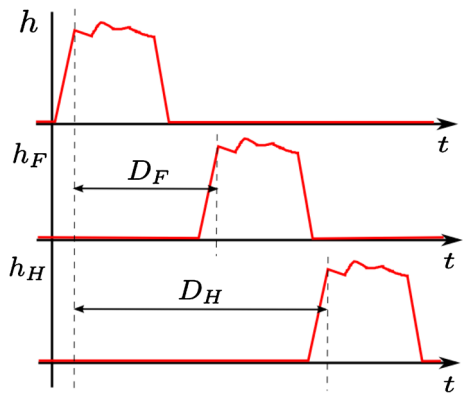


Fig. 8 A sketch of how the recorded height history (*upper plot*) is shifted in time before being applied to front (*middle plot*) and hind legs (*lower plot*)

from the vision rate $F_s = 15$ Hz and the minimum forward velocity $V_{f_{\text{min}}} = 0.1$ m/s. As soon as the index of the buffer increases we find the maximum height values that have been stored in the past. The idea is to apply to the front legs the step height which was stored D_F seconds before. D_F is the time interval (delay) in which the robot covers the horizontal distance $P_x = 1.25$ m from the camera to the center of the height map area while trotting at velocity V_f (Fig. 5). This value is changed in real time if V_f is changing (e.g., if the robot slows down the D_F will increase). The delay D_H applied to the hind legs is higher to account for the fact that these legs are located further (by an additional distance of d_{FH}) from the obstacle compared to the front legs, according to the robot's direction of motion (see Fig. 8).

$$D_F = \frac{P_x}{V_f + \delta}, \quad D_H = \frac{P_x + d_{FH}}{V_f + \delta} \tag{1}$$

where δ is a small constant to prevent division by zero issues. After the variable delays D_i are computed (at each control cycle), the indexes I_i associated to them are obtained as follows:

$$I_i = \text{floor}(F_s D_i) \quad i = F, H \tag{2}$$

Then the values of the modifications to the step heights h_F and h_H for the front and hind legs, respectively, are extracted from the buffer (see Fig. 7) to be added to the nominal step height. A first-order digital filter is implemented to smooth the step-wise discontinuities in $h_i(t)$ signals since the vision rate (15 Hz) is lower than the control rate (1 kHz).

The velocity can also be changed according to the difficulty of the obstacle. For instance, when facing higher obstacles the robot slows down and it speeds up when crossing lower obstacles. It is important to note that, to have a proper estimate of the distance covered by the robot, it is preferable to keep the velocity constant while overcoming the obstacle.

The proposed feature increases robustness and allows the robot to successfully trot over obstacles. When an obstacle above a certain threshold is detected, the robot's commanded forward velocity is set to 0. Mapping and path/foothold planning approaches can be used to overcome such difficulties. The implementations of those more sophisticated obstacle avoidance/climbing strategies are part of future work.

4 Locomotion control and vision

We showed in the previous part that the visual system can send to the reactive controller framework (RCF) [3] a qualitative localization and information about the terrain to adapt autonomously and continuously the trotting parameters. It this part we will explain how the RCF is using the visual feedback to perform closed-loop gait adjustment. The structure of the RCF consists of two main blocks, named motion generation and motion control blocks (see lower part of Fig. 9), that work in harmony to provide suitable feet trajectory and to control the trunk motion and posture.

The robot locomotion is obtained using a motion generation algorithm based on central pattern generators (CPG), which are neural networks responsible for generating gait patterns [14]. Our CPGs are emulated by four non-linear oscillators, synchronized according to the desired gait, which provide outputs as position references for each foot. Each oscillator has parameters directly associated to the step height H_s, step length L_s, step frequency f_s, forward velocity V_f and duty factor D_f, which we consider as locomotion parameters that can be modified independently. This modulation

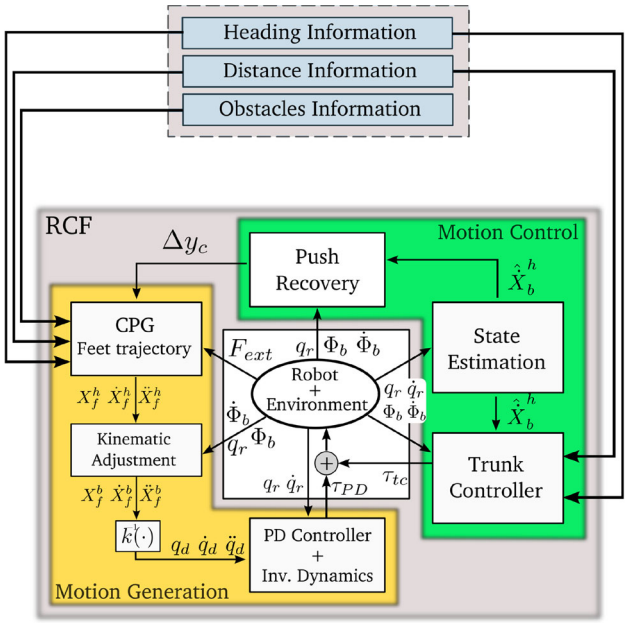


Fig. 9 Coupling between the vision process information and the reactive controller framework (RCF). The vision block, in *blue*, provides spatial information to the motion generation and motion control blocks (color figure online)

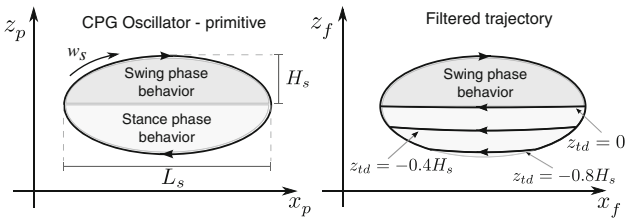


Fig. 10 The foot trajectory generated by the CPG oscillator (on the *left*) and the trajectory modulated by the non-linear filter (on the *right*), expressed in the robot's base frame. z_p and x_p are the reference coordinates of the primitive's trajectory, while z_f and x_f are the filtered references sent to the joint controller. The primitive has variable angular frequency w_s modulated according to f_s and D_f. z_{td} is the filter parameter which determines where the original elliptic trajectory has to be interrupted

allows to govern the robot using these parameters as control inputs that can be adjusted according to terrain irregularities, obstacle heights and target tracking errors.

The oscillator's output is a primitive that has an elliptical shape determined by the step length and step height, as depicted in Fig. 10 on the left. The primitive is modulated by a non-linear filter according to a relative distance named step depth $z_{td} \in [-H_s, H_s]$ that is acquired at the foot touchdown (Fig. 10, right). The non-linear filter modulation increases the locomotion robustness by adapting the primitive to irregular surfaces.

During the stance phase of the legs, the non-linear filters impose feet references to achieve robot omnidirectional locomotion with motion constraint satisfaction. These references

are relative motions between the feet and the torso computed according to the desired linear and angular velocities for the torso. In this paper, the robot locomotion is determined by means of the desired forward velocity V_f and a desired angular yaw velocity $\dot{\psi}_d$ for the torso.

The robot balance is controlled by the motion control block that is composed mainly of a push recovery and a trunk controller algorithm. The push recovery algorithm computes suitable footholds that drive the robot naturally to the default posture after an external disturbance. The trunk controller algorithm computes the joint torque references of the stance legs, to obtain a desired force and moment acting on the trunk.

In principle, the RCF is an approach designed to improve the locomotion robustness on irregular and unknown terrains. In this paper we fuse vision processing information with the RCF to make decisions and provide a spatial reference to the robot. The coupling between the RCF and the vision processing algorithm is depicted in Fig. 9.

The vision process sends information to two main algorithms: the CPG and the trunk controller. As in the CPG algorithm each locomotion parameter can be independently modulated, we introduce the idea of considering each locomotion parameter as a control input and use the vision information to generate control actions to modulate them, e.g.:

- Step height: directly proportional to the obstacles' height,
- Forward velocity: inversely proportional to the "degree of terrain irregularity" or directly proportional to the distance error to the tracked target,
- Robot turning: directly proportional to the angular error to the tracked target,
- Duty factor: directly proportional to the "degree of terrain irregularity".

The "degree of terrain irregularity" has been simply defined as the variance of the height obstacles values buffered. The visual data sent to the CPG block are the highest obstacle height and its relative distance and robot heading deviation from the target object. The heading information is used to control the robot turning and the distance information is used to control the robot's forward velocity. We have implemented proportional control actions, described as follows:

$$\dot{\psi}_d = -K_{p_\psi}\psi_h \tag{3}$$

$$V_f = K_{p_v}(P_0 - P_{\text{target}}) \tag{4}$$

where $\dot{\psi}_d$ and V_f are the desired turning velocity and desired forward velocity, respectively. The vision process provides the heading angle ψ_h and the target distance P_{target}. The parameters K_{p_ψ} and K_{p_v} are controller gains. P_0 is the desired distance from the target.

This visual feedback contributes substantially to the locomotion robustness by providing a qualitative localization and informations about the terrain. Such knowledge allows the robot to adjust each step height to overcome obstacles. A suitable step height is crucial to reduce the risk of foot-object frontal impacts and also important to reduce energy consumption during the leg swing phase.

To be coherent with the RCF concept, the vision process also sends information to the trunk controller about the tracked target distance and heading deviation. Both control laws described in (3) and (4) are considered as references. Then, the trunk controller computes joint torques to apply forces and moments according to V_f and $\dot{\psi}_d$ errors, i.e.:

$$F_{V_f} = K_f(V_f - \dot{x}_b^h) \tag{5}$$

$$M_\psi = K_m(\dot{\psi}_d - \dot{\psi}) \tag{6}$$

where F_{V_f} and M_ψ are, respectively, the force and the moment applied to the trunk to reduce motion errors. The actual forward velocity is denoted by $\dot{x}_b^h$ and the actual robot turning by $\dot{\psi}$. The parameters K_f and K_m are controller gains.

5 Experiments

Those algorithms have been experimentally tested indoors and outdoors on our quadruped robot. To demonstrate the performance and robustness of our system we ran two kinds of experiments that are explained after a short description of our research platform.

5.1 Our platform: HyQ robot

The experimental platform used in this study is the quadruped robot HyQ [21,22], (see Fig. 1). It is a hydraulically actuated machine that weighs 85 kg, is 1 m long and has upper and lower leg segment lengths of 0.35 m. The robot's legs have three degrees of freedom each, two hydraulic joints in the sagittal plane (hip and knee flexion/extension) and another for hip abduction/adduction. Each joint has 120° range of motion and is controllable in torque and position. The maximum joint torque is 145 Nm for the hydraulic. Semini et al. [22] describe HyQ's design and specifications in detail.

Since 2011, HyQ has demonstrated a wide range of static and dynamic motions such as a crawl gait, walking trot over flat, inclined and rough terrain (indoors and outdoors), flying trot, squat jumps, rearing, balancing under disturbances and step reflexes [3,5,9,22–24].

5.2 Indoor experiments on a treadmill

In the performed indoor experiments the robot is trotting on a treadmill while tracking a colored target (mounted on the crane in front of the robot). The robot velocity is modified to

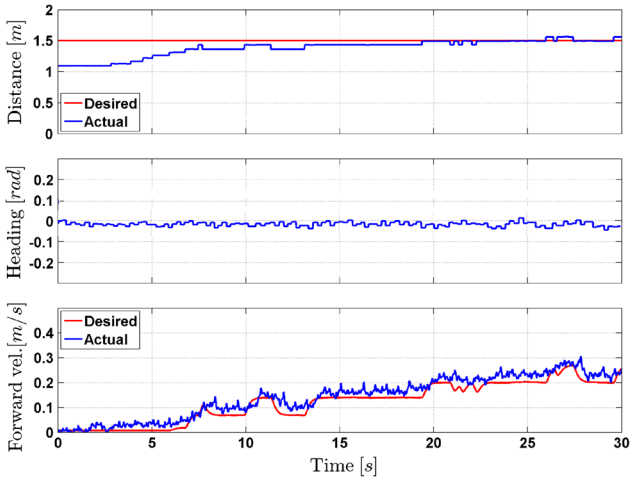


Fig. 11 Results of indoor experiment. *Top* actual (*blue*) and the desired (*red*) distance to the object; *middle* relative heading angle; *bottom* actual (*blue*) and desired (*red*) forward velocity of the robot (color figure online)

keep the desired distance even in presence of external disturbances. If an external operator changes the treadmill velocity, the robot adapts its velocity accordingly to keep the desired distance. At the same time, the control of the heading corrects autonomously any lateral drift in the locomotion direction and helps to keep the robot in the middle of the treadmill. This experiment shows the effectiveness of the static tracking to keep the robot on the treadmill autonomously.

Figure 11 shows the correction of the relative heading angle and the modification of the robot forward speed according to the vision feedback. The top plot displays the actual (blue) and the desired (red) (1.5 m) distance to the object. The middle plot shows the actual (blue) and desired (red) relative heading angle.The bottom plot illustrates the forward velocity.

As an extension for this experiment it is possible to set a moving target instead of a static object. In this case the robot is able for example to follow a "leader" (at a desired distance) that is walking in front of the robot.

Without the heading and distance control the robot occasionally drifted to one side, for reasons such as unbalanced weight, inaccuracies in the model, calibration errors or external forces. Sometimes it was also turning while moving over big obstacles placed on the treadmill or when someone was pushing it. During those experiments an operator had to pull the robot back to the center of the treadmill with slings when it was getting too close to the lateral limits of the treadmill.

The addition of visual feedback to the controller allows the robot to keep its position on the moving treadmill autonomously: when the trot in place is started and the tracked object is in sight, the system does not need any further intervention from the user. HyQ keeps the object in sight by turning right and left and keeping the distance to the object constant for randomly changing treadmill speeds between 0 and 0.3 m/s.

5.3 Outdoor experiments

Outdoor experiments demonstrate the robot's capability to trot towards a target object while overcoming obstacles placed in its way on a 10 m track, see Figs. 12, 13 and 14. In this particular case, the vision is used for heading control (targeting an object lying on the ground at the end of the track) and for obstacle detection but the distance control was disabled. In these experiments we want to show how vision can enhance locomotion by adapting the CPG step height to different variable delay digital buffer presented in Sect. 3.4. The experiment was repeated for different situations (flat terrain, flat terrain with pieces of wood, rough terrain with rocks lower than 10 cm), under different lighting conditions (artificial light, natural light day/evening), and with unavoidable

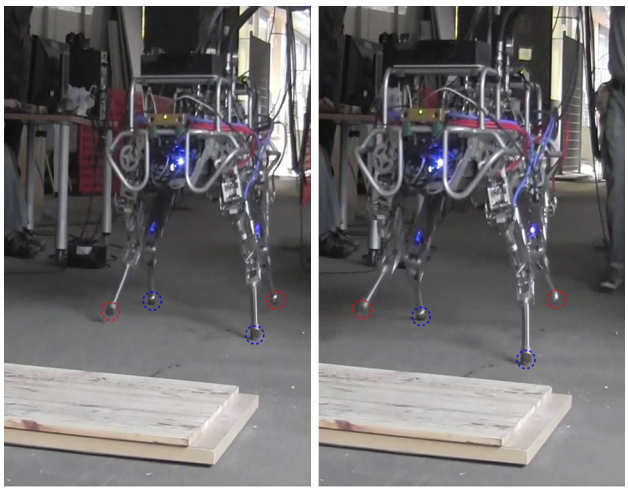


Fig. 12 Outdoor experiments. **a** At 1 m distance to the obstacle with the default step height: 7 cm. **b** At 50 cm distance to the obstacle the step height has been increased to 11 cm. The obstacle height is 4 cm. Feet in the *blue dashed circles* are in stance while the *red dashed circles* highlight the feet that are swinging (color figure online)

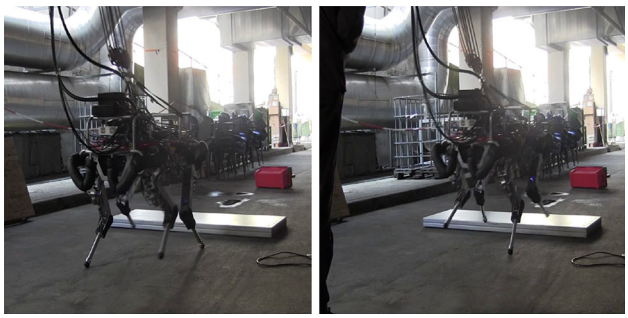


Fig. 13 First outdoor experiment to test the step height modification and heading control while passing over a step of 7.5 cm

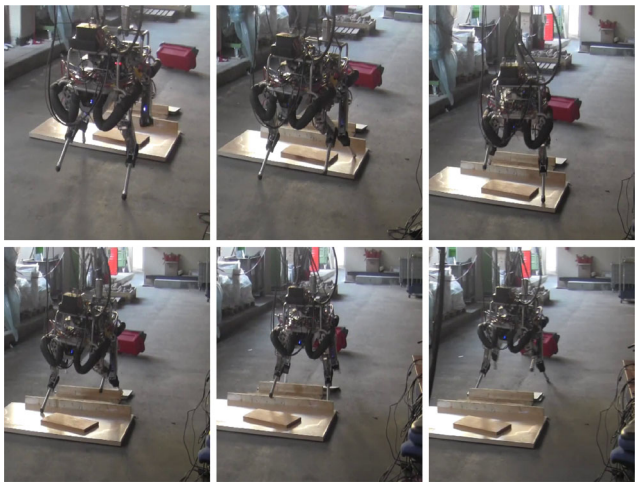


Fig. 14 Second outdoor experiment to test the step height modification and heading control on rough terrain. During this experiment the step height was set to the highest one (20 cm), the obstacle height is around 9.5 cm. Note that the robot was swaying to the *left* and *right* during this experiment, but it was correcting the heading by tracking the *red box* in front of it (color figure online)

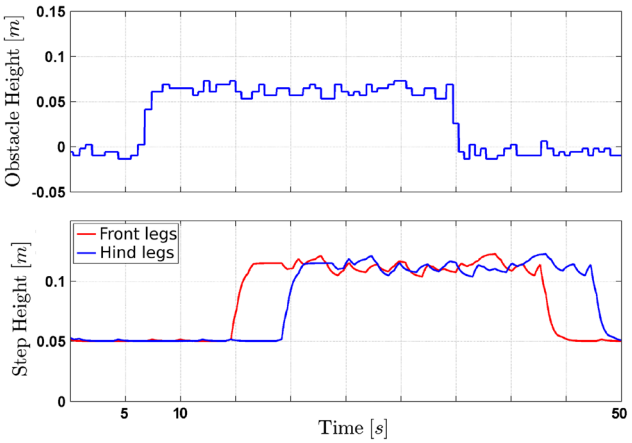


Fig. 15 Outdoor experiments, step height adaptation. *Top* detected maximum obstacle height; *bottom* controlled step height for the front (*red*) and hind (*blue*) legs (color figure online)

obstacles (big rocks, people crossing). The controller was in this case modifying the direction and the step height according to the obstacles detected in front of it.

Figures 15 and 16 show, respectively, the step height modification according to the obstacles detected and the heading control without distance control. Figure 12 shows the obstacle used for this experiment, which were pieces of wood piled up on the track. The average height of the obstacles was around 10–12 cm. As the obstacles are detected (at 1.5 m from the robot) a delay is introduced (proportional to the robot velocity) before modifying the step height of the front legs. This allows to obtain the step height required to overcome the obstacle only at the moment in which the robot is passing over it and not earlier, as described in Sect. 3.4.

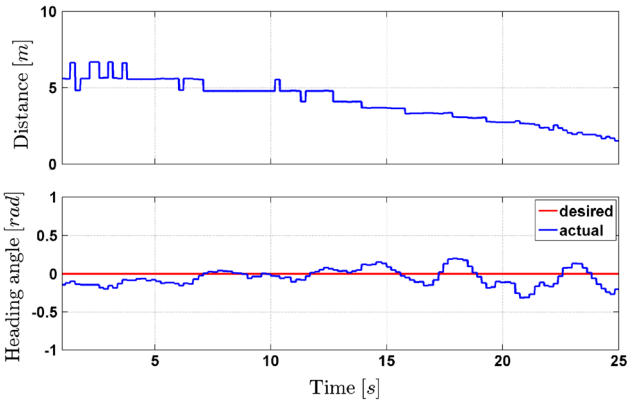


Fig. 16 Outdoor experiments, heading control. *Top* distance to the target; *bottom* actual (*blue*) and desired (*red*) relative heading angle (color figure online)

6 Discussion

The results of this paper showed the heading control, the distance control and the step height adjustment. The heading control and the distance control are robust since the tracking works robustly. Despite of noise in the signals sent by the vision process, the robot behavior is smooth. In the rare case that the tracker is lost (e.g., during fast motions or occlusions), the robot stops at its current position and a new target has to be selected by the operator. It has to be noted that the occurrence of a lost tracker can be reduced by fusing the color information with shape-based processing.

Also, the noise in the obstacle height estimation can lead to undesired behavior. The robot can sometimes miss the obstacle in front of it, due to an underestimation of the obstacle height, or stop if the value is overestimated due to noise.

It has to be mentioned that the rough terrain in this study is achieved by randomly putting obstacles on the flat ground (pieces of wood, rocks), while the robot is secured by a harness connected to a rail to prevent damage to the robot in case it stumbles or falls. The rope that connects the harness to the rail is hanging loosely while the robot is trotting.

A first limitation of the approach can come from the fact that the method has been developed for straight line locomotion and will need modifications to allow curved trajectories. To solve this, a solution is to mount the camera on a pan and tilt unit to allow to have a look to the terrain before turning and overwrite the obstacles height buffer with the new values obtained.

Another limitation of our approach is that in certain cases a strong foot-object frontal impact can occur, which prevents the robot from overcoming the obstacle even if the step height was sufficient. To solve this, we plan in the future to combine our method with the step reflex behavior that we have recently published in [9].

7 Conclusion and future work

In this paper, we presented a new reactive controller using position, force, inertial measurements and vision data for closed-loop gait adjustment. The achieved result is a significant step towards rendering HyQ more autonomous. We show that high-level information from perception sensors is now available to perform closed-loop gait adjustment. Results show that without any mapping or planning we achieved autonomous trotting on rough terrain. The robot is capable of navigating in a straight line towards a visual goal and reach it while correcting for drift or compensating for disturbances. Furthermore, the earlier presented reactive locomotion framework has been improved, as obstacles can now be detected and the robot can autonomously slow down and stop without requiring swift intervention by a human operator.

In future work we aim to extend this work by adding gait transitions. For example when the terrain becomes very rough the robot can slow down and locomote with a static gait instead of trotting. On the vision side, we plan to perform state estimation and 3D mapping that can be used for foothold planning in such very rough terrains.

Acknowledgments This work is an extended version of a previously published paper at the TEPRA conference [4]. The research has been funded by the Fondazione Istituto Italiano di Tecnologia. Jonas Buchli is supported by a Swiss National Science Foundation professorship. Successful experiments on HyQ are the fruit of many people's contributions during the last years. For a full list of lab members please visit www.iit.it/hyq.

References

1. Artner NM (2008) A comparison of mean shift tracking methods. In: 12th Central European Seminar on computer graphics, pp 197–204
2. Bajracharya M, Ma J, Malchano M, Perkins A, Rizzi A, Matthies L (2013) High fidelity day/night stereo mapping with vegetation and negative obstacle detection for vision-in-the-loop walking. In: Proceedings of the IEEE/RSJ International Conference on intelligent robots and systems (IROS), pp 3663–3670
3. Barasuol V, Buchli J, Semini C, Frigerio M, De Pieri ER, Caldwell DG (2013) A reactive controller framework for quadrupedal locomotion on challenging terrain. In: 2013 IEEE International Conference on robotics and automation (ICRA)
4. Bazeille S, Barasuol V, Focchi M, Havoutis I, Frigerio M, Buchli J, Semini C, Caldwell DG (2013) Vision enhanced reactive locomotion control for trotting on rough terrain. In: IEEE International Conference on technologies for practical robot applications (TePRA)
5. Boaventura T, Semini C, Buchli J, Frigerio M, Focchi M, Caldwell DG (2012) Dynamic torque control of a hydraulic quadruped robot. In: Proceedings of the IEEE Int. Conference on robotics and automation (ICRA)
6. Bradski G (1998) Computer video face tracking for use in a perceptual user interface. Intel Technol J 1:1–15
7. Chilian A, Hirschmuller H (2009) Stereo camera based navigation of mobile robots on rough terrain. In: Intelligent robots and systems (IROS) IEEE/RSJ International Conference on, pp 4571–4576
8. Filitchkin P, Byl K (2012) Feature-based terrain classification for littledog. In: Intelligent robots and systems (IROS), 2012 IEEE/RSJ International Conference on, pp 1387–1392
9. Focchi M, Barasuol V, Havoutis I, Buchli J, Semini C, Caldwell DG (2013) Local reflex generation for obstacle negotiation in quadrupedal locomotion. In: Int. Conf. on climbing and walking robots (CLAWAR)
10. Fujita M, Kitano H (1998) Development of an autonomous quadruped robot for robot entertainment. Auton Robots 5(1):7–18
11. Hartley R, Zisserman A (2000) Multiple view geometry in computer vision, vol 2. Cambridge Univ Press, Cambridge
12. Havoutis I, Semini C, Buchli J, Caldwell DG (2013) Quadrupedal trotting with active compliance. IEEE International Conference on mechatronics (ICM)
13. Howard A (2008) Real-time stereo visual odometry for autonomous ground vehicles. In: Proceedings of the 2008 IEEE/RSJ International Conference on intelligent robots and systems (IROS), pp 3946–3952
14. Ijspeert AJ (2008) 2008 special issue: central pattern generators for locomotion control in animals and robots: a review. Neural Netw 21(4):642–653
15. Kalakrishnan M, Buchli J, Pastor P, Mistry M, Schaal S (2011) Learning, planning, and control for quadruped locomotion over challenging terrain. Int J Robot Res 30(2):236–258
16. Kolter JZ, Youngjun K, Ng AY (2009) Stereo vision and terrain modeling for quadruped robots. In: IEEE International Conference robotics and automation (ICRA) on, pp 1557–1564
17. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Computer Vis 60(2):91–110
18. Mühlmann K, Maier D, Hesser J, Männer R (2002) Calculating dense disparity maps from color stereo images, an efficient implementation. Int J Computer Vis 47(1–3):79–88
19. Murray D, Little JJ (2000) Using real-time stereo vision for mobile robot navigation. Auton Robots 8(2):161–171
20. Pippine J, Hackett D, Watson A (2011) An overview of the defense advanced research projects agency's learning locomotion program. Int J Robot Res 30:141–144
21. Semini C (2010) HyQ—design and development of a hydraulically actuated quadruped robot. Ph.D. thesis, Italian Institute of Technology and University of Genoa
22. Semini C, Tsagarakis NG, Guglielmino E, Focchi M, Cannella F, Caldwell DG (2011) Design of HyQ—a hydraulically and electrically actuated quadruped robot. J Syst Control Eng 225(6):831–849
23. Semini C, Khan H, Frigerio M, Boaventura T, Focchi M, Buchli J, Caldwell DG (2012) Design and scaling of versatile quadruped robots. In: Int. Conf. on climbing and walking robots (CLAWAR)
24. Semini C, Barasuol V, Boaventura T, Frigerio M, Buchli J (2013) Is active impedance the key to a breakthrough for legged robots? In: International Symposium on robotics research (ISRR)
25. Shao X, Yang Y, Wang W (2012) Obstacle crossing with stereo vision for a quadruped robot. In: Mechatronics and automation (ICMA), 2012 International Conference on, pp 1738–1743
26. Zhou H, Yuan Y, Shi C (2009) Object tracking using sift features and mean shift. Computer Vis Image Underst 113(3):345–352