Vision Enhanced Reactive Locomotion Control for Trotting on Rough Terrain

Stéphane Bazeille*, Victor Barasuol†, Michele Focchi*, Ioannis Havoutis*, Marco Frigerio*, Jonas Buchli‡,
Claudio Semini*, Darwin G. Caldwell*

* Dep. of Advanced Robotics,
Istituto Italiano di Tecnologia (IIT),
via Morego, 30, 16163 Genova
<first name>.<last name>@iit.it

†Dep. of Automation and Systems,
Federal University of Santa Catarina (UFSC),
Florianpolis, SC, Brazil 88040-970
victor@das.ufsc.br

‡Agile & Dexterous Robotics Lab,
ETH Zurich,
Tannenstr. 3, 8092 Zürich
buchlij@ethz.ch

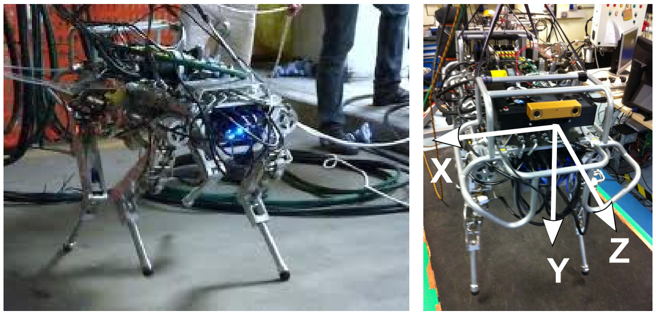
***Abstract*—Legged robots have the potential to navigate in more challenging terrain than wheeled robots do. Unfortunately, their control is more difficult because they have to deal with the traditional mapping and path planning problems, as well as foothold computation, leg trajectories and posture control in order to achieve successful navigation. Many parameters need to be adjusted in real time to keep the robot stable and safe while it is moving. In this paper, we will present a new framework for a quadruped robot, which performs goal-oriented navigation on unknown rough terrain by using inertial measurement data and stereo vision. This framework includes perception and control, and allows the robot to navigate in a straight line forward to a visual goal in a difficult environment. The developed rough terrain locomotion system does not need any mapping or path planning: the stereo camera is used to visually guide the robot and evaluate the terrain roughness and an inertial measurement unit (IMU) is used for posture control. This new framework is an important step forward to achieve fully autonomous navigation because in the case of problems in the SLAM mapping, a reactive locomotion controller is always active. This ensures stable locomotion in rough terrain, by combining direct visual feedback and inertial measurements. By implementing this controller, an autonomous navigation system has been developed, which is goal-oriented and overcomes disturbances from the ground, the robot weight, or external forces. Indoor and outdoor experiments with our quadruped robot show the effectiveness and the robustness of this framework.**

***Index Terms*—Reactive walking, active compliance, goal oriented navigation, visual servoing, quadruped robot.**

I. 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 and electric actuation developed at the Department of Advanced Robotics at the Italian Institute of Technology (IIT) [1]. HyQ is fast, robust, actively compliant and built for dynamic locomotion.

Our previous work focused on dynamic locomotion, mainly trotting, using active compliance and low-level feedback coming directly from the on-board inertial measurement unit (IMU) for stabilization [2], [3]. Such low-level control can reliably negotiate flat and rough terrain while following intuitive high-level feedback, i.e. desired velocity and desired heading, from an operator. This provides a solid foundation for building up a set of higher-level controllers that deal with the cognitive aspects of locomotion and navigation, and further increase HyQ's autonomy. Along this line we added a stereo vision sensor 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, path planning.



(a)

(b)

Fig. 1. Pictures of IIT's quadruped robot HyQ. (a) outdoors, without stereo camera; (b) indoors, with stereo camera, showing the definition of the camera coordinate frame.

In this paper we present an extension to the previously presented reactive controller [2], that additionally uses visual information to guide the low-level trotting controller towards a goal or over rough terrain. Visual feedback is crucial in such scenarios as open-loop approaches, e.g. dead-reckoning, quickly accumulate errors due to foot slippage, a non-uniform weight distribution, slopes, terrain irregularites or due to physical disturbances.

The vision system feeds-back the target's position and the height and distance of the obstacles that are in front of the robot. This in turn influences the step height of the feet of the quadruped while it also regulates the robot's forward velocity according to the roughness of the terrain during locomotion.

In this work we focus on a trotting behaviour without mapping or path planning. With the integration of a stereo-camera-based vision system, we are able to autonomously trot to a given target, track a target while traversing challenging terrain and also in presence of strong external disturbances. Adding perception to the control framework allows us to compensate for possible drift due to foot slippage or transient loss of balance. Furthermore, it allows us to make the behaviour safer by detecting obstacles, slowing down when necessary, increasing the duty factor or the step height of the trot and in case of an obstacle that cannot be overcome, stop.

Contribution: A new reactive controller using position, force and inertial measurements in combination with vision data. This controller allows to perform a fully autonomous reactive trot in an unknown terrain.

Contents: The structure of the paper is organized as follows. In Section II, we present a review of related work on quadruped robot navigation, in Section III, we provide details about our perception algorithms. Section IV describes the controller of our robot. In Section V we present our quadruped robot and the results of indoor and outdoor experiments. Finally, in Section VI we conclude and discuss future work.

II. Related work

Quadruped locomotion is an active area of research. However, up to now few people have worked on the integration of vision sensors on quadruped 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, Kalakrishnan *et al.* used pre-scanned maps and a marker-based tracking system on *LittleDog* [4]. On the other han,d the authors in [5] presented a framework for terrain modelling and pose estimation without a-priori information on the environment, using a stereo camera. Filitchkin and Byl used a monocular camera to do terrain classification and in turn influence the locomotion behaviour of their *LittleDog* quadruped [6]. The authors of [7] performed position estimation and terrain modelling using a stereo camera, while Shao *et al.* [8] presented obstacle avoidance using a stereo vision-based terrain modelling approach. Howard in [9] introduced a state estimation approach that combines a number of different data sets from stereo camera, IMU, odometry and GPS to aim at long-term position accuracy.

III. Environment Perception

As mentioned in Section I, 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, by making decisions about the robot inside a given environment. To perceive the environment and improve the locomotion robustness of the system we therefore added a stereo camera to the robot.

Our camera is a *Bumblebee2* firewire colour camera from *Point Grey*. It provides a focal length of 2.5mm, a field of view of 97 degrees, a maximum resolution of 1024 x 768 at 20 fps, a 12cm baseline, and it is pre-calibrated against distortions and misalignment. A 640x480 pixel disparity map was computed at 5Hz on our system.

A. Colour tracking and depth computation for heading and distance control

For colour tracking we are using a modified CAMShift algorithm [10]. 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 [11]. In this method, the kernel is a simple step function applied to a colour probability map. The colour probability of each pixel is computed using a method called *histogram back projection*. The algorithm creates a confidence map in the new image based on the colour 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.

Colour is represented as Hue from the HSV colour model, a colour space that is more consistent than the standard RGB colour space under changing illumination. Since Hue is unstable at low saturation, the colour histograms do not include pixels with saturation below a threshold.

Similarly, the minimum and maximum intensity values were neglected to skip pixels that are very bright or very dark.

A few processing methods have also been used to improve the tracking under similar conditions to the ones presented here. As a quadruped robot's trunk is moving a lot during locomotion, we increased the search region and we post-processed the back projection image using morphological filters to remove noise and keep the tracker on the target object. In Fig. 2 we show the detection of a red object indoors. The tracking was done using the left camera with a frame rate of 5Hz. The implemented tracking has been tried on many different objects and under different conditions. For example, we were usually manually selecting a random object present in the scene as goal for the robot.

As we explained previously we use a stereo camera to obtain two different views of the scene. By comparing the images, the relative depth information can be obtained in the form of a disparity map, which is inversely proportional to the differences in distance to the objects. We used this to extract the 3D position of the object.

To obtain this 3D information several steps are required (Fig. 3):

- The images are first undistorted to ensure that the observed images are purely projectional.

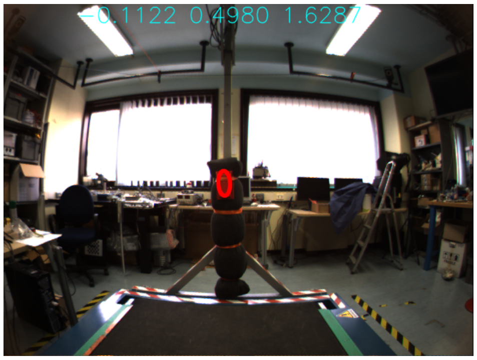
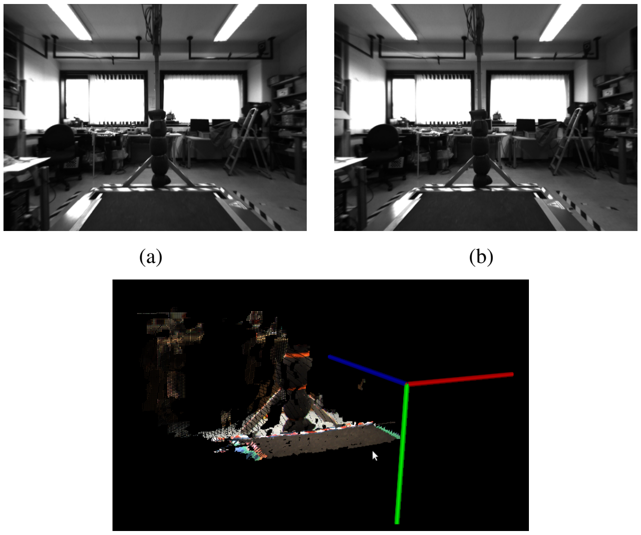


Fig. 2. Example of colour detection of an object. This object is tracked continuously and its 3D position (expressed in meters in the camera frame defined in 3(c)) is displayed in blue in the top of the image.


(a)

(b)

(c)

Fig. 3. (a) (b) Left and right rectified images during an indoor experiment. (c) Associated point cloud computed from the stereo pair. We show on this point cloud display, the camera reference frame: x in red, y in green and z in blue.

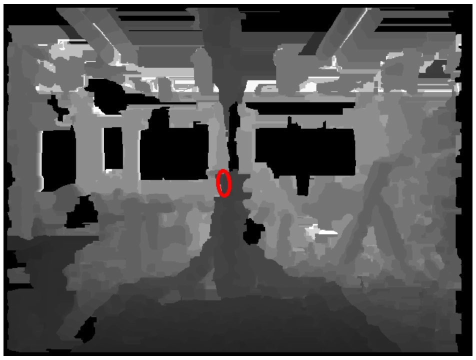
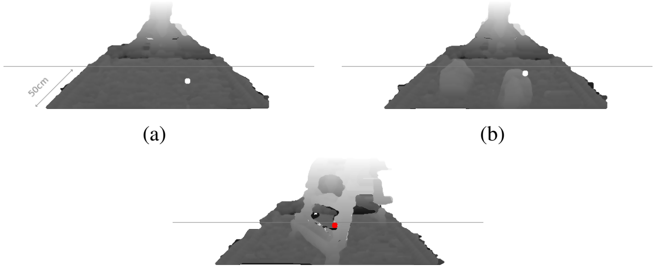


Fig. 4. Depth map showing the tracked coloured object with a red oval in the center of the image.



(a)

(b)

(c)

Fig. 5. Example of height map in the robot frame. (a) Without obstacles; (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 respectivly 0.70, 0.63, 0.20. The last obstacle is too big to be crossed.

- The images are rectified to project them back to a common plane to allow comparison of the image pairs,
- The disparity map is extracted by computing the displacement of relative features in the left and right images,
- The disparity is then post-processed to fill holes by interpolating missing values,
- The 3D information is extracted from each pixel by using disparity information and the perspective model.

Given this 3D information, we can extract the object in the colour image (chosen as the gravity center of the tracking ellipse) and we extract x and y position in the camera reference frame. To be robust on the computation of z the affected value is chosen as the median depth of all the pixel in the ellipse in the depth map (Fig. 4 shows the depth map).

The computed 3D point is then compared to the previous value to keep the consistency and remove outliers. In case of outliers we send the previous position value to the controller to keep the position update rate constant. Robustness of this computation is discussed in Section V.

B. *Visual obstacle detection for step height adjustment.*

As we have computed the 3D point cloud of the environment we have a height map. We will use it to detect obstacles and modify the forward velocity and the step height. To do so, we look for the highest obstacle in a small area in front of the robot. If the obstacle can be crossed (according to robot leg maximum retraction capability this means to be lower than 25cm) the step height is modified accordingly. Otherwise we stop the robot. 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 gate in this study. Fig. 5 shows the computed height map. This height map is constrained to be 2m large at a distance between 1m and 1.5m. The grey line represents the 1.5m distance. The highest point is computed in this 2m x 0.5m box. To avoid outliers the map is temporally filtered over three images. The map accuracy is discussed in Section V.

The computed maximum height of the obstacle and its distance to the camera is sent to the controller and the step height is modified accordingly with a function *variable delay digital buffer*, which delays the modification of this parameter depending on the distance of this obstacle and the robot speed. An additional delay is considered for the hind legs to account for the fact that they are located further from the obstacle than the front legs (in the direction of motion). This increases robustness and allows the robot to successfully trot over obstacles. When a too-big obstacle is detected, the robot's forward velocity is set to 0. The implementation of more sophisticated obstacle avoidance strategies is part of future work.

C. *Controller input*

The vision system provides visual data with a frame rate of 5Hz to the robot controller. The visual data packet contains the 3D position vector of the tracked object and the height and the distance to the highest obstacle in front of the robot. These values are expressed in the camera reference frame and are properly translated into the robot base frame via an appropriate homogeneous transform. Values are temporally filtered before being sent to the robot controller. The values of the height after being processed through the *variable delay digital buffer* are filtered again before they are used in the controller. The accuracy of the obstacle height is about $\pm 2cm$ and $\pm 5cm$ for the distance.

IV. Locomotion Control and Vision

The robot is able to track targets due to a combination of vision-based tracking with our Reactive Controller Framework (RCF) [2]. The structure of the RCF consists of two main blocks, named *Motion Generation* and *Motion Control* blocks (see lower part of Fig. 6), that work in harmony to provide suitable feet trajectory and to control the trunk motion and posture.

The robot locomotion is obtained by using a motion generation algorithm based on Central Pattern Generators (CPG), which are neural networks responsible for generating animals gait patterns [12]. Our CPGs are emulated by four non-linear oscillators, synchronized according to the desired gait, that provide outputs as position references for each foot. Each oscillator has parameters directly associated to the step height, step length, forward velocity and duty factor, which we consider as *locomotion parameters* that can be modulated independently. This modulation allows to govern the robot by using these parameters as control inputs that can be adjusted according to, e.g., terrain irregularities, obstacle heights and target tracking errors.

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. 6.

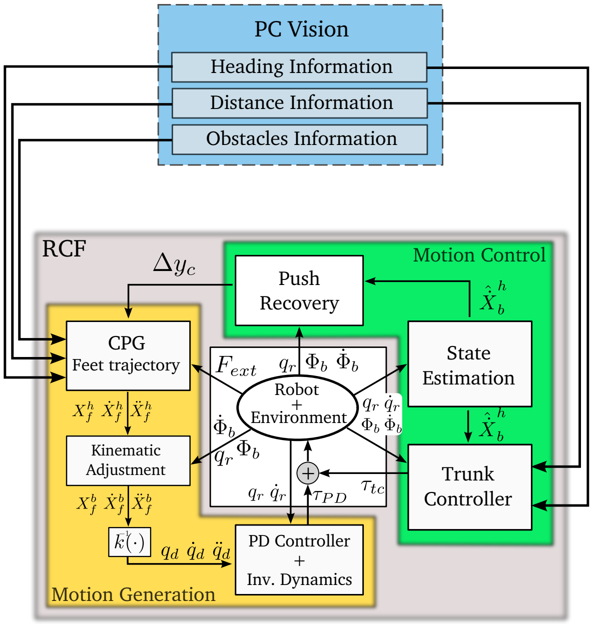


Fig. 6. Coupling between the vision process information and the Reactive Controller Framework (RCF). The Vision block, in blue, provides spatial information to the motion generation and motion control blocks. The RCF diagram is extracted from [2].

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.

In this paper's approach, the vision process sends information to the CPG block about terrain irregularity and relative distance and robot heading deviation from a certain object. The heading information is used to control the robot turning and the distance information is used to control the robot's forward velocity. For simplicity we have implemented simple proportional control actions, described as follows:

$$\dot{\psi}_d = -K_{p_\psi} \psi_h \tag{1}$$

$$V_f = K_{p_v} (P_0 - P_{target}) \tag{2}$$

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.

The vision process contributes substantially to the locomotion robustness by providing information about the terrain. Such knowledge allows 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 (1) and (2) 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.:

$$F_{V_f} = K_f(V_f - \dot{x}_b^h) \quad (3)$$

$$M_\psi = K_m(\dot{\psi}_d - \dot{\psi}) \quad (4)$$

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.

V. Experiments

As previously explained, those algorithms have been experimentally tested indoors and outdoors on our quadruped robot.

A. Our platform: HyQ Robot

The experimental platform used in this study is the versatile quadruped robot HyQ [1], [13], Fig. 1. It is a hydraulically and electrically actuated machine that weighs $70kg$, is $1m$ long and has upper and lower leg segment lengths of $0.35m$. The robot's legs have three degrees of freedom each, two hydraulic joints in the sagittal plane (hip and knee flexion/extension) and one electric joint for hip adduction/abduction. Each joint has 120° range of motion and is controllable in torque and position. The maximum joint torque is 145Nm for the hydraulic and 152Nm for the electric joints. Semini *et al.* [1] describe HyQ's design and specifications in detail.

To demonstrate the performance and robustness of our system we ran two kinds of experiments with a static coloured object.

B. Indoor experiment on a treadmill

In the performed indoor experiments the robot is trotting on a treadmill while tracking a coloured target and keeping a desired distance from it. The robot velocity is modified to keep the desired distance in spite of external disturbances. If an external operator changes the treadmill velocity the robot adapts its velocity accordingly to track 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. Results are shows in Fig. 7. As an extension for this experiment it is possible to set a moving target instead of a static object. In this case the robot will be able for example to follow a ”leader”.

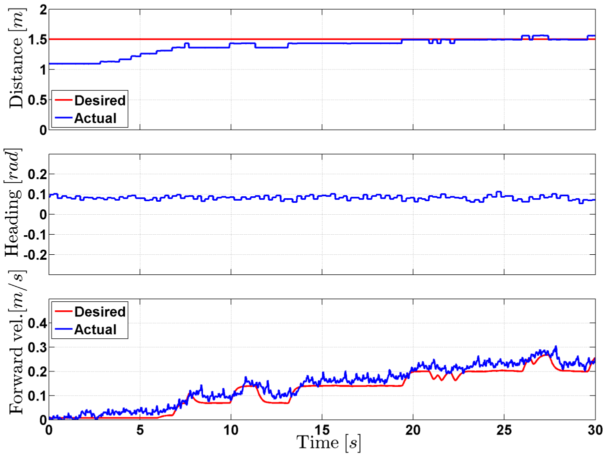


Fig. 7. Results of indoor experiment. Top: actual (blue) and the desired (red) distance to the object; Middle: relative heading angle; Bottom: actual (blue) and desired (red) forward velocity of the robot.

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 control the robot with slings when it was getting close to the lateral limits of the treadmill.

The addition of visual feedback to the controller made the robot completely autonomous now: 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 $0.3m/s$.

Fig. 7 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) $(1.5m)$ distance to the object. The middle plot shows the actual (blue) and desired (red) relative heading angle. The bottom plot illustrates the forward velocity.

C. Outdoor experiments

These experiments demonstrate the robot's capability to trot towards a target object while overcoming the obstacles placed in its way on a 10m track. In this particular case, the vision is used for heading control and step height adjustment, the distance control was disabled since this experiment's goal was to reach the target object lying on the ground at the end of the track. The experiment was repeated for different situations (flat terrain, flat terrain with pieces of wood, rough terrain with rocks lower than 10cm), under different lightning conditions, and with unavoidable 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.

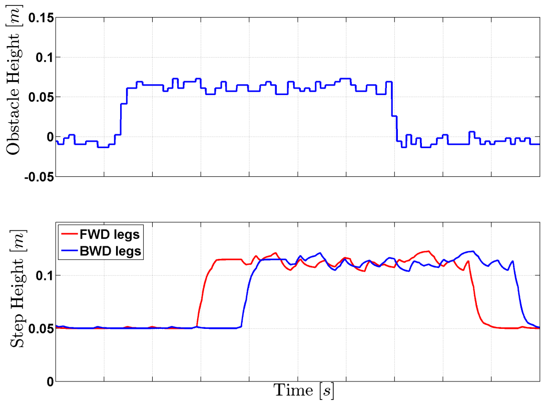


Fig. 8. Outdoor experiments, step height adaptation. Top: detected maximum obstacle height; Bottom: controlled step height for the forward (red) and backward (blue) legs.

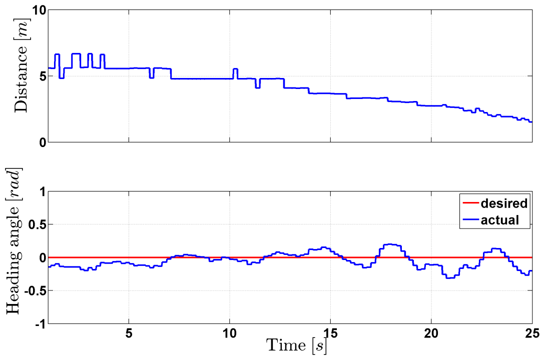


Fig. 9. Outdoor experiments, heading control. Top: distance to the target; Bottom: actual (blue) and desired (red) relative heading angle.

Fig. 8 and 9 show respectively the step height modification according to the obstacles detected and the heading control without distance control. The obstacle used for this experiment were pieces of wood piled up on the track. Average height was around 6-7cm. As the obstacle is detected (at $1.5m$ 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 the step height required to overcome the obstacle only at the moment in which the robot is approaching it and not before. A longer delay is present for the hind legs that are negotiating the obstacle after the front legs.

D. Discussion on the results

Results shows the heading control, the distance control and the step height adjustment. The heading control and the distance control are robust since the tracking works correctly. Despite noise in the signals sent by the vision process, the robot behaviour is smooth. In the rare case that the tracker is lost (e.g. during fast motions), the robot goes out of the track and the robot stops. This can be improved by fusing the colour information with shape-based processing or adding a second tracked object for example.

The noise in the step height adjustment can lead to errors, as the robot can sometimes miss the obstacle crossing due to an underestimation of the value or stop if the value is overestimated.

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), as the robot is currently secured by a harness connected to a rail to prevent it from falling.

VI. Conclusion and Future Work

In this paper, we presented our latest progress of the HyQ project. The achieved result is a significant step towards rendering HyQ autonomous, as we show that high-level information from perception sensors are now available to the locomotion controller. 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 is improved and safer now, as obstacles can be detected and the robot can stop by itself without any operator intervention.

In the future, we will extend this work by adding gait transitions, for example by slowing down and walk instead of trotting or changing from forward to backward motion. On the vision side, we plan to perform state estimation and 3D mapping for foothold planning.

References

[1] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. G. Caldwell, "Design of HyQ - a hydraulically and electrically actuated quadruped robot," *Journal of Systems and Control Engineering*, vol. 225, no. 6, pp. 831–849, 2011.

[2] V. Barasuol, J. Buchli, C. Semini, M. Frigerio, E. R. De Pieri, and D. G. Caldwell, "A reactive controller framework for quadrupedal locomotion on challenging terrain," in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, 2013, (accepted).

[3] I. Havoutis, C. Semini, J. Buchli, and D. G. Caldwell, "Quadrupedal trotting with active compliance," *IEEE International Conference on Mechatronics (ICM)*, 2013, (accepted).

[4] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, "Learning, Planning, and Control for Quadruped Locomotion over Challenging Terrain," *Int. J. of Robotics Research*, vol. 30, pp. 236–258, 2011.

[5] J. Z. Kolter, K. Youngjun, and A. Y. Ng, "Stereo vision and terrain modeling for quadruped robots," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, 2009, pp. 1557 –1564.

[6] P. Filitchkin and K. Byl, "Feature-based terrain classification for littledog," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, oct. 2012, pp. 1387 –1392.

[7] A. Chilian and H. Hirschmuller, "Stereo camera based navigation of mobile robots on rough terrain," in *Intelligent Robots and Systems(IROS) IEEE/RSJ International Conference on*, 2009, pp. 4571 –4576.

[8] X. Shao, Y. Yang, and W. Wang, "Obstacle crossing with stereo vision for a quadruped robot," in *Mechatronics and Automation (ICMA), 2012 International Conference on*, aug. 2012, pp. 1738 –1743.

[9] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," in *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008, pp. 3946–3952.

[10] G. Bradski, "Computer video face tracking for use in a perceptual user interface," *Intel Technology Journal*, 1998.

[11] N. M. Artner, "A comparison of mean shift tracking methods," in *12th Central European Seminar on Computer Graphics*, 2008, pp. 197–204.

[12] A. J. Ijspeert, "2008 special issue: Central pattern generators for locomotion control in animals and robots: A review," *Neural Netw.*, vol. 21, no. 4, pp. 642–653, May 2008.

[13] C. Semini, "HyQ – design and development of a hydraulically actuated quadruped robot," Ph.D. dissertation, Italian Institute of Technology and University of Genoa, 2010.