



HAL
open science

High-slope Terrain Locomotion for Torque-Controlled Quadruped Robots

Michele Focchi, Andrea del Prete, Ioannis Havoutis, Roy Featherstone,
Darwin Caldwell, Claudio Semini

► **To cite this version:**

Michele Focchi, Andrea del Prete, Ioannis Havoutis, Roy Featherstone, Darwin Caldwell, et al.. High-slope Terrain Locomotion for Torque-Controlled Quadruped Robots. 2015. hal-01137225v1

HAL Id: hal-01137225

<https://hal.science/hal-01137225v1>

Preprint submitted on 30 Mar 2015 (v1), last revised 20 Nov 2018 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Autonomous Robots

High-slope Terrain Locomotion for Torque-Controlled Quadruped Robots

--Manuscript Draft--

Manuscript Number:	AURO-D-14-00144
Full Title:	High-slope Terrain Locomotion for Torque-Controlled Quadruped Robots
Article Type:	S.I. : Whole-body control for Humanoid Robots
Keywords:	whole-body control; locomotion; quadrupeds; torque control; force control
Corresponding Author:	Andrea Del Prete, Ph.D. LAAS-CNRS Toulouse, Haute Garonne FRANCE
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	LAAS-CNRS
Corresponding Author's Secondary Institution:	
First Author:	Michele Focchi, PhD
First Author Secondary Information:	
Order of Authors:	Michele Focchi, PhD Andrea Del Prete, Ph.D. Ioannis Havoutis, PhD Roy Featherstone, Prof. Darwin G. Caldwell, Prof. Claudio Semini, PhD
Order of Authors Secondary Information:	
Abstract:	<p>Research into legged robotics is primarily motivated by the prospects of building machines that are able to navigate in challenging and complex environments that are predominantly non-flat. In this context, control of contact forces is fundamental to ensure stable contacts and stability of the robot. In this paper we propose a planning/control framework for quasi-static walking of quadrupedal robots, implemented for a demanding application in which regulation of ground reaction forces is crucial. Experimental results demonstrate that our 75-kg quadruped robot is able to walk inside two high-slope (50°) V-shaped walls; an achievement that to the authors' best knowledge has never been presented before. Furthermore, the robot is distributing its weight among the stance legs so as to optimize user-defined criteria. We compute joint torques that result in no foot slippage, fulfillment of the unilateral constraints of the contact forces and minimization of the actuators effort. This paper presents an experimental study that compares the proposed framework with different state-of-the-art control strategies, demonstrating the effectiveness and robustness of our approach.</p>
Suggested Reviewers:	Jerry Pratt IHMC jpratt@ihmc.us Expert of the field Johannes Engelsberger DLR Johannes.Engelsberger@dlr.de He worked on locomotion Sang-Ho Hyon

	College of Science and Technology, Ritsumeikan University sangho@ieee.org He worked on torque-controlled robots.
	Michael Mistry University of Birmingham m.n.mistry@bham.ac.uk He worked on locomotion control of quadruped robots.

High-slope Terrain Locomotion for Torque-Controlled Quadruped Robots

Michele Focchi* · Andrea del Prete⁺ · Ioannis Havoutis* · Roy Featherstone* · Darwin G. Caldwell* · Claudio Semini*

Received: date / Accepted: date

Abstract Research into legged robotics is primarily motivated by the prospects of building machines that are able to navigate in challenging and complex environments that are predominantly non-flat. In this context, control of contact forces is fundamental to ensure stable contacts and stability of the robot. In this paper we propose a planning/control framework for quasi-static walking of quadrupedal robots, implemented for a demanding application in which regulation of ground reaction forces is crucial. Experimental results demonstrate that our 75-kg quadruped robot is able to walk inside two high-slope (50°) V-shaped walls; an achievement that to the authors' best knowledge has never been presented before. Furthermore, the robot is distributing its weight among the stance legs so as to optimize user-defined criteria. We compute joint torques that result in no foot slippage, fulfillment of the unilateral constraints of the contact forces and minimization of the actuators effort. This paper presents an experimental study that compares the proposed framework with different state-of-the-art control strategies, demonstrating the effectiveness and robustness of our approach.

Keywords Whole-body control · Locomotion

1 Introduction

Current research on legged robots is motivated by their potential impact in real-world scenarios such as disaster recovery scenes. Such environments require systems capable of robustly negotiating uneven and sloped terrains. In recent years the field has seen remarkable advances in the theoretical tools, which have allowed legged robots to tackle challenging and possibly dynamic tasks in simulation [14, 12]. However, to this date, experimental results have been limited to few platforms and tasks, still not matching the complexity of the real world. Righetti et al. [20] experimented with

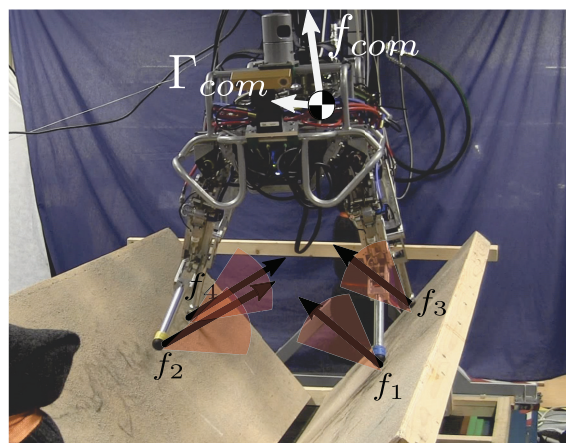


Fig. 1: HyQ quadruped robot walking inside a 50°-inclined groove. Desired wrench (force, moments) at the CoM is depicted in white. Ground reaction forces are in brown friction while cone constraints are indicated in shaded red. The wall inclination is θ .

walking up a slope of 26° with the *Little Dog* quadruped robot. On the quadruped robot StarLETH [6] Hutter et al. [8] used a contact-force optimization method to achieve static walking on a surface with approximately 40° inclination. Regarding contact force control in humanoid robots, so far research has mainly focused on balancing experiments on flat ground [9, 17, 23].

This substantial gap between simulation and reality is due to a number of different factors. The lack of high-fidelity joint torque control is probably the first difficulty [8, 4, 2]. Moreover, the identification of inertial and geometric parameters of these high-DoF multibody systems is usually cumbersome [15], and errors in the identified dynamical models introduce unknown disturbances in the control actions. Furthermore, the estimation of the system state is typically a

complex procedure that merges multiple sensor data in order to exploit all the available information [1].

The contribution of this work is to tackle the complete set of these issues, combining different ideas from planning to control, and applying them to a challenging test case. We present experimental results on a 75-kg torque-controlled quadruped robot, showing that it can walk in between two high-slope (50°) V-shaped walls (Fig. 1).

To the best of our knowledge this is the first implementation of such a task on a real robot. Such a scenario is the most challenging for testing the capabilities of our controller, because it allows a greater slope inclination, and therefore requires a greater rotation of the ground reaction forces (GRFs) compared to walking on a flat or inclined surface. For instance, a static walk on a single slope of 50° would not be possible with a friction coefficient of 1 or less. Nonetheless, our approach is applicable to any kind of sloped terrains.

We also present a comparison with other state-of-the-art controllers, which demonstrates the importance, for this kind of task, of accurately controlling the contact forces. Fig. 2 presents the building blocks of our control framework. The *motion control* block is a whole-body controller inspired by the works presented in [17, 6]. The approach is rather general, because it can deal with any number of contacts, as long as normal directions and friction coefficients are known or estimated. The robot can adapt to uneven surfaces while distributing its weight over the supporting contact points in an *optimal* manner. Moreover, the method does not require contact force measurements and avoids joint torque discontinuities. The *motion generation* block computes desired trajectories for the CoM, the base orientation and the swing foot to achieve a static walking pattern. The latter adapts to the geometry of the terrain to achieve a stable foothold and to ensure physical feasibility (e.g. not to violate the constraints of the stance feet). The rest of this work is organized as follows: Section 2 and Section 3 describe the controller implementation and the *motion generation*, respectively. Section 4 introduces our robotic platform and reports the experimental results obtained, along with the values used for all the parameters of the algorithm. Section 5 discusses some practical issues that are often overlooked when working in simulation, namely joint torque limits, model identification and friction estimation. Finally, Section 6 draws the conclusions and presents future work directions.

2 Whole body controller with optimization of ground reaction forces

This section describes the control architecture developed for quadrupedal robot walking on inclined terrain. The controller computes desired joint torques, that are tracked by the low-level torque controllers [2]. Our objectives are to regulate

i) the position of the center of mass (CoM) and ii) the orientation of the base of the robot. We do this by computing Ground Reaction Forces (GRFs) at the stance feet that result in the desired i) acceleration of the CoM and ii) angular acceleration of the robot's base. At the same time, we take into account the constraints imposed by the friction cones.

2.1 Centroidal robot dynamics

Following the results presented in [16], the centroidal robot dynamics can be described as:

$$\dot{l} = m(\dot{x}_{com} + g) = F_{com} \quad (1)$$

$$\dot{h} = I_G \dot{\omega}_G + \dot{I}_G \omega_G = \Gamma_{com}, \quad (2)$$

where \dot{l} and \dot{h} are the rate of change of linear and angular momentum respectively, $g \in \mathbb{R}^3$ is the gravity acceleration vector, $m \in \mathbb{R}$ is the total robot's mass, $I_G \in \mathbb{R}^{3 \times 3}$ is the centroidal rotational inertia, $\dot{x}_{com} \in \mathbb{R}^3$ is the acceleration of the CoM, $\dot{\omega}_G \in \mathbb{R}^3$ is the rotational acceleration of an equivalent rigid body with the inertia I_G , and finally $F_{com} \in \mathbb{R}^3$ and $\Gamma_{com} \in \mathbb{R}^3$ are the net external force and moment at the CoM, respectively.

The design of the controller is based on the following assumptions. First, we assume that $\dot{I}_G \omega_G \simeq 0$: this is reasonable because in our experiments the robot moves slowly. Second, since most of the robot's mass is located in its base (i.e. 47 out of 75 kg), we approximate the CoM (x_{com}) and the average angular velocity of the whole robot (ω_G) with the CoM of the trunk $x_{com-base}$ ¹ and the angular velocity of the base ω_b . Third, since our platform has nearly point-like feet, we assume that it cannot generate moments at the contacts. Fourth, we assume that the GRFs are the only external forces acting on the system. Under these assumptions, we can rewrite (1) and (2) expressing the net force and moment at the CoM as functions of the c GRFs (i.e. $f_1, \dots, f_c \in \mathbb{R}^3$, where c is the number of stance feet):

$$m(\ddot{x}_{com} + g) = \sum_{i=1}^c f_i \quad (3)$$

$$I_G \dot{\omega}_b \simeq \sum_{i=1}^c (f_i \times p_{com,i}), \quad (4)$$

where $p_{com,i} \in \mathbb{R}^3$ is a vector going from the CoM to the position of the i^{th} foot defined in an inertial world frame \mathcal{W} (see Fig. 3). These two equations are the base of our control design because they describe how the GRFs affect the acceleration of the CoM and the angular acceleration of the robot's base. We now design two proportional-derivative control laws to compute the desired values of \ddot{x}_{com} and $\dot{\omega}_b$. Then, we will find the GRFs that allow us to achieve these desired accelerations.

¹ In the following we keep using x_{com} even if in the implementation we actually used $x_{com-base}$.

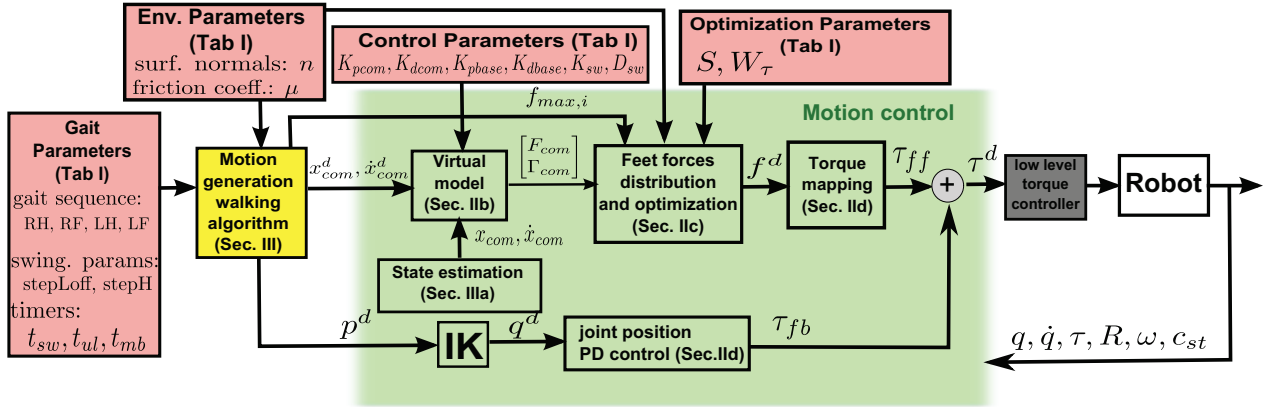


Fig. 2: Block diagram of our framework. The motion generation block (yellow) computes the input trajectories for CoM and joints, while the motion control block (green) computes the reference torques for the low-level controller (grey). Light red blocks indicate user-defined input parameters. Each block is detailed in the sections indicated in parenthesis.

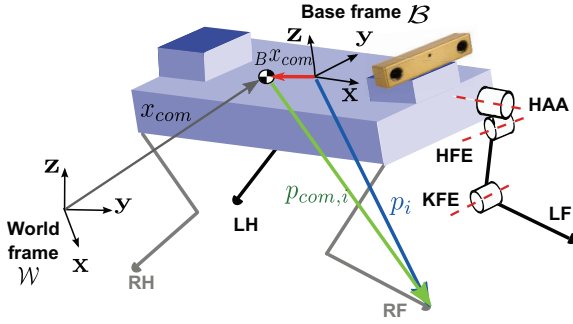


Fig. 3: Summary of the nomenclature used in the paper. Leg labels: left front(LF), right front(RF), left hind (LH) and right hind(RH). The world frame \mathcal{W} ; the base frame \mathcal{B} (attached to the geometric center of the robot body). Left subscripts indicate the reference frame, for instance Bx_{com} is the location of the CoM w.r.t. the base frame. In case of no left subscript, quantities are expressed w.r.t. \mathcal{W} .

2.2 Control of CoM's position and base's orientation

We compute the desired acceleration of the CoM $\ddot{x}_{com}^d \in \mathbb{R}^3$ using a PD control law:

$$\ddot{x}_{com}^d = K_{pcom}(x_{com}^d - x_{com}) + K_{dcom}(\dot{x}_{com}^d - \dot{x}_{com}), \quad (5)$$

where $x_{com}^d \in \mathbb{R}^3$ is the desired position of the CoM, whereas $K_{pcom} \in \mathbb{R}^{3 \times 3}$ and $K_{dcom} \in \mathbb{R}^{3 \times 3}$ are positive-definite diagonal matrices of proportional and derivative gains, respectively. Similarly, we compute the desired angular acceleration of the robot's base $\ddot{\omega}_b^d \in \mathbb{R}^3$ as:

$$\ddot{\omega}_b^d = K_{pbase}e(R_b^d R_b^\top) + K_{dbase}(\omega_b^d - \omega_b), \quad (6)$$

where $R_b \in \mathbb{R}^{3 \times 3}$ and $R_b^d \in \mathbb{R}^{3 \times 3}$ are rotation matrices representing the actual and desired orientation of the base w.r.t. the world reference frame, respectively, $e(\cdot) : \mathbb{R}^{3 \times 3} \rightarrow \mathbb{R}^3$ is

a mapping from a rotation matrix to the associated rotation vector, $\omega_b \in \mathbb{R}^3$ is the angular velocity of the base, whereas $K_{pbase} \in \mathbb{R}^{3 \times 3}$ and $K_{dbase} \in \mathbb{R}^{3 \times 3}$ are positive-definite diagonal matrices of proportional and derivative gains, respectively.

2.3 Computation of the desired GRFs

Given a desired value of the acceleration of the CoM and the angular acceleration of the robot's base, we want to compute the desired GRFs f . We rewrite (3) and (4) in matrix form as:

$$\underbrace{\begin{bmatrix} I & \dots & I \\ [p_{com,1} \times] & \dots & [p_{com,c} \times] \end{bmatrix}}_A \underbrace{\begin{bmatrix} f_1 \\ \vdots \\ f_c \end{bmatrix}}_f = \underbrace{\begin{bmatrix} m(\ddot{x}_{com}^d + g) \\ I_g \ddot{\omega}_b^d \end{bmatrix}}_b, \quad (7)$$

where we replaced the actual accelerations with the desired accelerations. This system has 6 equations and $k = 3c$ unknowns; since in our experiments $3 \leq c \leq 4$, typically the system has infinite solutions. We exploit this redundancy to ensure the respect of the inequality constraints imposed by the friction cones. At every control loop we solve the following quadratic program:

$$\begin{aligned} f^d &= \underset{f \in \mathbb{R}^k}{\operatorname{argmin}} (Af - b)^\top S (Af - b) + \alpha f^\top W f \\ \text{s.t. } \underline{d} &< Cf < \bar{d}, \end{aligned} \quad (8)$$

where $S \in \mathbb{R}^{6 \times 6}$ and $W \in \mathbb{R}^{k \times k}$ are positive-definite weight matrices, $\alpha \in \mathbb{R}$ weighs the secondary objective, $C \in \mathbb{R}^{p \times k}$ is the inequality constraint matrix, $\underline{d}, \bar{d} \in \mathbb{R}^p$ the lower/upper bound respectively, with p being the number of inequality constraints. These ensure that i) the GRFs lie inside the friction cones and ii) the normal components of the GRFs stay within some user-defined values.

We approximate friction cones with square pyramids to express them with linear constraints. We then define C , \underline{d} and \bar{d} as:

$$C = \begin{bmatrix} C_0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & C_c \end{bmatrix} \quad \underline{d} = \begin{bmatrix} \underline{d}_0 \\ \vdots \\ \underline{d}_c \end{bmatrix} \quad \bar{d} = \begin{bmatrix} \bar{d}_0 \\ \vdots \\ \bar{d}_c \end{bmatrix}, \quad (9)$$

with:

$$C_i = \begin{bmatrix} (-\mu_i n_i + t_{1_i})^\top \\ (-\mu_i n_i + t_{2_i})^\top \\ (\mu_i n_i + t_{2_i})^\top \\ (\mu_i n_i + t_{1_i})^\top \\ n_i^\top \end{bmatrix} \quad \underline{d}_i = \begin{bmatrix} -\infty \\ -\infty \\ 0 \\ 0 \\ f_{\min_i} \end{bmatrix} \quad \bar{d}_i = \begin{bmatrix} 0 \\ 0 \\ \infty \\ \infty \\ f_{\max_i} \end{bmatrix}, \quad (10)$$

where $n_i \in \mathbb{R}^3$ is the direction normal to the surface, $t_{1_i}, t_{2_i} \in \mathbb{R}^3$ are the tangential directions, $\mu_i \in \mathbb{R}$ is the coefficient of friction, and $f_{\min_i}, f_{\max_i} \in \mathbb{R}$ are the minimum and maximum allowed values for the i^{th} normal force, respectively; all these values are of course relative to the i^{th} contact. In the cost function of (8) the term $f^\top W f$ regularizes the solution by trading-off the tracking of \ddot{x}_{com} and $\dot{\omega}_b$ with small-magnitude GRFs. We can use the weight matrix W to penalize certain force directions (e.g. to penalize tangential forces). Actually, in our experiments we found more useful to penalize high joint torques rather than high GRFs (see Section 5.3).

Remark 1 According to our robotic-platform specificities, the presented controller is sufficient to control the whole system. The robot has 18 DoFs (12 joints plus 6 DoFs of the floating base), but as long as it stands on four feet it is subject to 12 rigid-contact constraints. This leaves only 6 unconstrained DoFs, which are exactly the number of DoFs controlled by the presented method. When the robot stands on three feet it has instead 9 unconstrained DoFs: in this phase the 3 additional DoFs are compensated by the control of the position of the swinging foot. However, for systems with more DoFs (e.g. humanoid robots) it is necessary to control the remaining redundancy.

Remark 2 Although this paper focuses on quadruped locomotion, the presented method can accommodate for any number of contact points. For instance we could use virtual models [18] to generate virtual forces at the end-effectors to achieve motion-force tasks. In case of physical interaction, we have to incorporate the effect of the additional contact forces on the centroidal dynamics (i.e. on the vector b in (8)). This would enable to include manipulation tasks to physically interact with the environment.

Remark 3 The weights of the two conflicting terms in the objective function of (8) must be carefully tuned through the parameter α . A too strong regularization causes big tracking errors, thus negatively affecting the robot equilibrium.

Remark 4 Problem (8) always has a solution. Nonetheless, if the desired accelerations require GRFs that violate the inequality constraints, the controller does “the best that it can” in the least-squares sense. Therefore, it is crucial to plan trajectories that are coherent with friction constraints.

2.4 Mapping of GRFs to joint torques

We compute the desired joint torques $\tau^d \in \mathbb{R}^n$ (where n is the number of joints) by superimposing two control actions. First, mapping the desired GRFs f^d into joint space we get the feedforward torques τ_{ff} :

$$\tau_{ff} = -S J_c^\top f^d, \quad (11)$$

where $J_c \in \mathbb{R}^{k \times n+6}$ is the stacked Jacobian of the contact points and $S = [I_{n \times n} \ 0_{n \times 6}]$ is a selection matrix that selects the actuated DoFs. This same mapping was used in [17] and it is valid only for quasi-static motion.

The second part consists of a proportional-derivative (PD) joint-position controller with low gains, which on average contributed only to $\approx 18\%$ of τ^d . This second term is motivated by safety reasons — hydraulic actuators can generate fast and powerful movements — and it is also used to move the swing leg. During the swing motion we increase the PD gains of the swing leg joints to improve tracking capabilities. Overall, we compute the desired torques τ^d that we command to the underlying joint-torque controllers [2] as:

$$\tau^d = \tau_{ff} + PD(q^d, \dot{q}^d, c_{st}), \quad (12)$$

where $q^d \in \mathbb{R}^n, \dot{q}^d \in \mathbb{R}^n$ are the desired joint positions and velocities, respectively, and $c_{st} \in \mathbb{R}^4$ is the vector of boolean variables representing the stance condition of the legs.

3 Static-Walking Algorithm for Quadrupeds

Our static-walking algorithm is a sequential repetition of the following phases: *move CoM*, *unload leg*, *swing leg*, *load leg*. Each phase is a state of the state machine depicted in Fig. 4. The *gait sequence* that we used in our climbing experiments is an input parameter of the walking algorithm and it is described in the Appendix. We assume that the robot starts with all the four legs in contact with the terrain. A boolean flag c_{st} represents the contact state; this flag can be modified by both the walking algorithm and the environment, depending on the current walking phase.

In the *move-CoM* phase the robot moves its CoM inside the support triangle formed by the three legs opposite to the one that is about to swing (Section 3.1). This ensures static equilibrium when breaking the contact. A timer regulates the duration t_{mcom} of the *move-CoM* phase. Then the *unload* phase starts. During this phase the load on the swing leg is

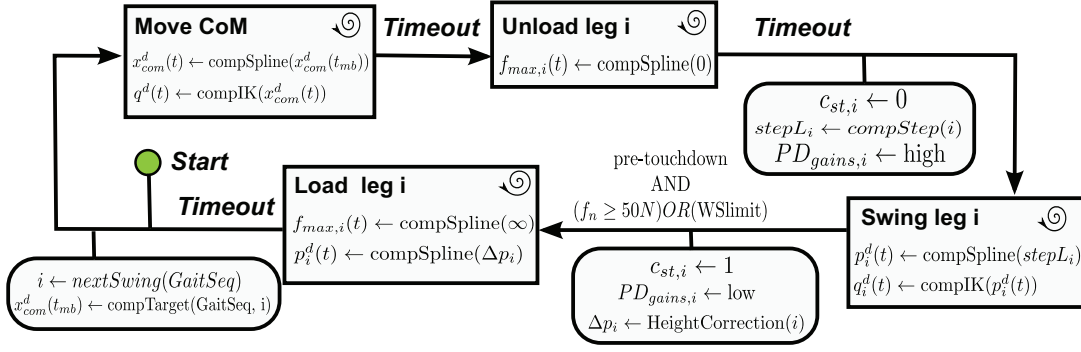


Fig. 4: Logic diagram of the state machine used in the static walking algorithm. Rectangles represent temporized phases and actions, arrows represent transitions, and rounded boxes represent actions associated to transitions.

gradually reduced to zero (Section 3.2). When the time t_{load} has elapsed, the *swing* phase begins with the computation of the desired foot placement for the swing foot (Section 3.3). Initially the foot swings away from the surface to achieve step clearance and then toward it (see Fig. 5). If in the pre-touchdown motion the ground is reached earlier than predicted the swing phase terminates. Otherwise the leg keeps moving (*searching motion*, see Section 3.3) until the foot makes contact. Finally, during the *load* phase, the number of stance legs is reset to four and the previous swing leg is gradually loaded, redistributing the weight equally on all the legs. At the same time the robot’s height is corrected (see Section 3.4). After the *load* phase the next swing leg is taken from the gait sequence and the cycle repeats. The input parameters for the static-walking algorithm (Fig. 2) are: the normals to the surface n_i at each contact point, the gait sequence $GaitS$, the step-length offset $stepLoff$, the step height $stepH$, and the time duration of each phase (t_{mcom} , t_{load} , t_{sw}) (see Tab. 1).

3.1 CoM’s Trajectory Generation

We estimate the CoM position x_{com} w.r.t. an inertial frame \mathcal{W} through leg odometry [13]. To do this we use joint-angle measurements and the model of the robot kinematics; under the assumption that the stance feet do not move (i.e. no slip), and given that there are always at least three stance feet, the position/orientation of the robot can always be uniquely determined.

In the *move CoM* phase the desired CoM trajectory is generated as a 5th-order minimum-jerk spline. The trajectory starts from the current CoM position ($x_{com}^d(0)$) and it ends at the target CoM $x_{com}^d(t_{mcom})$. The target CoM is computed so that $P_{xy}x_{com}^d(t_{mcom})$ lies inside the next support triangle T , where $P_{xy} \in \mathbb{R}^{3 \times 3}$ is a projector into a plane perpendicular to gravity (see Fig. 5). Since the steps are adapted to the terrain geometry during the walking, the support triangle can change its inclination w.r.t. gravity, because the feet

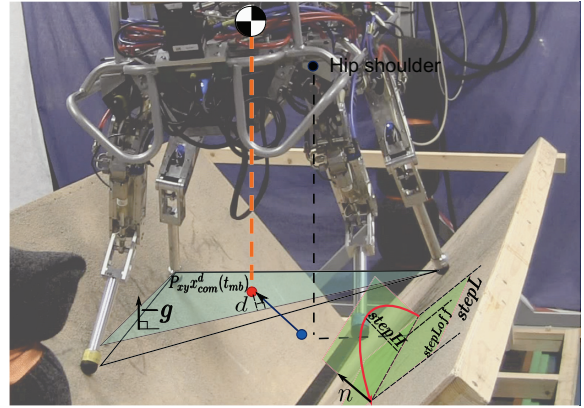


Fig. 5: HyQ robot walking inside a 50°-inclined groove. CoM is depicted in black and white. The wall inclination is θ . The red dot represents the projection $P_{xy}x_{com}^d(t_{mcom})$ of the desired CoM position $x_{com}^d(t_{mcom})$ on the *stable convex hull* (light blue), which is a projection of the support triangle (black) on a plane orthogonal to gravity. The desired trajectory of the swing leg lies on a plane normal to the ground surface, and it depends on the step height ($stepH$) and length ($stepL$).

may not be at the same height. Therefore, to ensure static equilibrium, we consider a projection of the triangle $P_{xy}T$. The position of $P_{xy}x_{com}^d(t_{mcom})$ inside $P_{xy}T$ can be tuned by changing the parameter d , which is the distance from the midpoint of the largest edge of the triangle. The smaller d , the smaller the static-equilibrium margin, but the bigger the walking velocity, because the amplitude of backward motions is reduced [3].

While we generate the desired trajectory of the CoM, we also need to compute the desired trajectories for the joint-level PD controllers of the stance legs. These joint trajectories must of course prevent the PD controllers from “fighting” against the whole-body controller. Since the legs of the robot have only 3 DoFs, we can analytically compute the

1 joints' trajectories from the feet's trajectories. The trajectories
 2 of the feet can in turn be computed from the desired
 3 CoM and base's orientation. In the following the left sub-
 4 script indicates the frame in which vectors are represented.
 5 Assuming that the stance feet do not move w.r.t. the inertial
 6 frame \mathcal{W} (i.e. ${}_w\dot{p}_i = 0, \forall i \in \text{StanceFeet}$), we compute the
 7 velocity of the i^{th} foot w.r.t. the base frame \mathcal{B} (i.e. ${}_b\dot{p}_i$) as a
 8 function of the CoM's velocity ${}_w\dot{x}_{com}$ and the base's angular
 9 velocity ${}_w\omega_b$:

$$12 \quad {}_b\dot{p}_i = R({}_w\dot{x}_{com} - [R^T {}_b p_i]_{\times} {}_w\omega_b), \quad (13)$$

15 where $R \in \mathbb{R}^{3 \times 3}$ is the rotation matrix from \mathcal{W} to \mathcal{B} . Using
 16 $\omega_b^d(t)$ and $\dot{x}_{com}^d(t)$ generated by the spliner (in world coordi-
 17 nates), we can then compute ${}_b p_i^d$ by integrating ${}_b\dot{p}_i^d$. Finally,
 18 we compute the desired joints' angles for each stance leg to
 19 use as references for the joint PD controllers.

23 3.2 Leg Loading/Unloading

26 The loading/unloading phases are fundamental to prevent
 27 discontinuities in the joints' torques any time that the num-
 28 ber of stance legs changes. We achieve the loading/unloading
 29 by splining the upper bound $f_{max,i}$ on the normal force of the
 30 leg i , from the current value to $10m/0$, where m is the mass
 31 of the robot. In particular, we update the \vec{d} vector (inequality
 32 constraints) at each time step during these phases.

36 3.3 Swing Leg

38 At the beginning of the *swing* phase we compute the step
 39 length $stepL_i$ as a fixed offset $stepL_{off}$ in the forward direc-
 40 tion w.r.t. the hip shoulder. Computing the footstep locations
 41 w.r.t. to the shoulder — rather than w.r.t. the actual foot po-
 42 sition — ensures no drift in the distance between the feet.
 43 Then the swing leg's trajectory $p_i^d(t)$ is generated on a plane
 44 normal to the ground's surface, as a function of the user-
 45 defined step height $stepH$ and step length $stepL$ (see Fig. 5).
 46 The first part of the swing motion is a spline through a via
 47 point to achieve step clearance; the second part consists of
 48 a surface approaching motion (*pre-touchdown*) towards the
 49 desired foot's placement. During the downward motion, if
 50 the contact is made before the planned foothold is reached,
 51 the leg stops. Conversely, if the step ends before making
 52 contact, the foot keeps moving at constant velocity along the
 53 ground's normal direction (*searching motion*) until it either
 54 makes contact or reaches the workspace's limits. The lowest
 55 singular value of the foot's Jacobian matrix is monitored to
 56 stop the leg motion before getting close to a singularity (e.g.
 57 leg completely extended).

3.4 Height Correction

Whenever the swing foot makes contact before/after expect-
 ed the foot-shoulder distance gets smaller/larger, and this
 affects the height of the robot. Thus, to prevent the robot
 from gradually “squatting”/“rising” during the walk, we cor-
 rect the leg's length. During the *load* phase, while changing
 the limit of the normal force, we also move the desired foot's
 position — and the relative desired joints' positions — of
 $\Delta p_i(Z)$:

$$\Delta p_i = - \left[h^d - e_3^\top {}_B x_{com} - (-e_3^\top p_i(t_{sw})) \right], \quad (14)$$

where $h^d \in \mathbb{R}$ is the desired robot's height computed at the
 CoM (see Tab. 1), ${}_B x_{com}$ is the position of the CoM in the
 frame \mathcal{B} (identified as explained in Section 5.2) and $e_3^\top \in \mathbb{R}^{1 \times 3}$
 is a vector selecting the z component.

4 Experiments

Before carrying out experiments on the real robot we exten-
 sively tested the framework in simulation with the SL soft-
 ware package [21]. (see attached video). However, for the
 sake of brevity, we report only the results obtained on the
 real robot.

4.1 HyQ Platform's Description

The experimental platform used in this work is a quadruped
 robot [22] (Fig. 5). The robot weighs 75 kg, it is $1m \times 0.5 \times$
 $1m$ ($L \times W \times H$) dimensions and it is equipped with 12 ac-
 tuated DoFs i.e. 3 DoFs for each leg. The hip abduction-
 adduction (*HAA*) joints (see Fig. 3) connect the legs to the
 robot's torso, creating the lateral leg's motion, while the hip
 and knee flexion/extension (*HFE* and *KFE*, respectively)
 create the motion in the sagittal plane. Linear hydraulic cylin-
 ders actuate the hip and knee flexion/extension (*HFE* and
KFE, respectively), while the *HAA* are rotary hydraulic ac-
 tuators. Load-cells, located at the end of the piston rods,
 measure the force of the hydraulic cylinders. By kinematic
 transformations, considering the lever arm between the pis-
 ton attachment and the joint axis, the joints' torques are
 computed. Similarly, a custom torque sensor, embedded in
 the *HAA* joint, provides direct measurements of the torque.
 An off-board pump brings the pressurized oil to the system
 through tethered hoses. An inertial measurement unit (IMU)
 provides measurements of orientation and angular velocity
 of the robot's base. Since most of the torque at the joints
 is due to the GRFs, we estimate the force at the i^{th} foot as:
 $f_i \simeq -J_i^{-\top} \tau_{leg_i}$, where $J_i \in \mathbb{R}^{3 \times 3}$ is the i^{th} leg's Jacobian and
 $\tau_{leg_i} \in \mathbb{R}^3$ are the torques of the i^{th} leg's joints. All the joints
 of the robot are torque controlled with a high-performance

low-level controller [2]. To verify the contact status of the feet we use a threshold on the normal component of the GRFs. The kinematic transformation used in this work are computed according to [5].

4.2 Groove

A good template to test the capability of our framework is the “horizontal groove” (see Fig.5). In this experiment the robot must actively push against the wall of the chimney to keep the GRFs inside the friction cones, so preventing slips and consequent falls. For practical reasons we built a horizontal chimney (*groove*) instead of a vertical one, which has been equivalently good for the proof of the concept. The robot has successfully walked through the entire length (2.5m) of the groove, with a wall inclination of $\theta = 50^\circ$. Before starting the controller the robot is already inside the groove, with all four feet in contact with the walls. A video of the experiments demonstrating multiple experimental trials of our control framework is enclosed.

4.2.1 Implementation Details

The control of the base’s orientation aims to maintain the robot’s trunk horizontal during the walk. Table 1 reports the values of the parameters used in the experiments. To be conservative we used a friction coefficient ($\mu = 0.5$) lower than the one that we estimated ($\mu = 1$) (see Section 5.1). This is important to improve the *robustness* w.r.t. the friction coefficient and terrain topology (i.e. inclination). Indeed, by using a conservative friction coefficient in the optimization problem, uncertainties in the estimation of the terrain’s normal direction are well tolerated. For example, in our experimental trials this ensured a tolerance to slope estimation errors of up to 18° .

The identification of the CoM’s position (see Section 5.2) was crucial for the success of the experiments. Despite having only 2.7cm of error (in the *xy* plane) w.r.t. the CoM computed from the CAD model, this was enough to make the robot fall after half a cycle.

The control loop for the low-level torque controller ran at 1 kHz, whereas the whole-body controller ran at 133 Hz. We solved the optimization problem (8) in real-time using the open-source software OOQP [7]. On the onboard pentium PC104 1GHz computer, running under a real-time Linux operating system, the resolution of (8) with $3c = 12$ variables ($c = 4$ contact points) and $5c = 20$ inequality constraints took on average 6.34 ms. We decided not to include the inequality constraints for the joints’ torques (see Section 5.3) because the 12 additional bilateral constraints increased the computation time to 9.82ms, thus exceeding the maximum time of 7.52ms.

Table 1: Parameters of the controller

Parameter	Symbol	Value
Wall inclination	θ [rad]	0.87
Friction coefficient	μ [1]	0.5
CoM proportional gain	K_{pcom} [N/m]	diag($10^3, 10^3, 500$)
CoM derivative gain	K_{dcom} [sN/m]	diag(200,200,0)
Attitude proportional gain	K_{pbase} [Nm/rad]	diag($10^3, 10^3, 10^3$)
Attitude derivative gain	K_{dbase} [sNm/rad]	diag(200,200,200)
Joint impedance control stiffness during swing	K_{sw} [Nm/rad]	300
Joint impedance control damping during swing	D_{sw} [sNm/rad]	6
Step length offset w.r.t. hip	$stepLoff$ [m]	0.11
Step height	$stepH$ [m]	0.1
Static stability margin	d [m]	0.09
Weights for CoM wrench components	S	diag(5, 5, 10, 10, 10, 10)
Weights for torque minimization (see 5.3)	W_τ	diag(5, 50, 2) 10^{-3}
Second objective weight	α	0.01
Gait sequence	$GaitS$	<i>RH,RF,LH,LF</i>
Phase durations	t_{mb}, t_{us}, t_{sw} [s]	4, 2.5, 2
Desired robot height	h^d [m]	0.6

4.2.2 Results

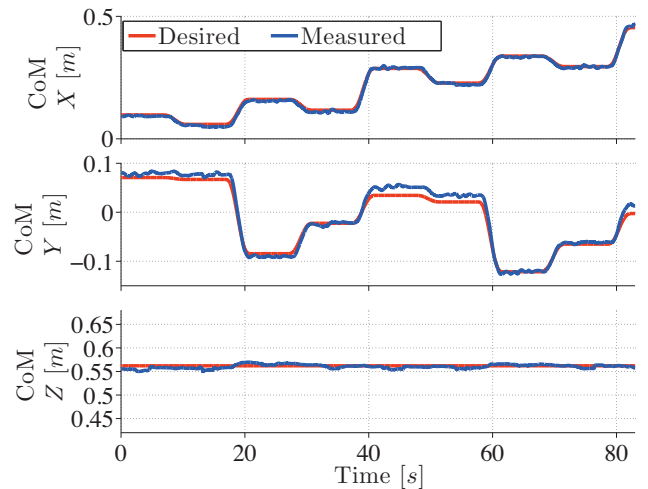


Fig. 6: Experimental results. Tracking of the center of mass during the walking.

Fig. 6 and 7 present the tracking of the CoM’s position and the base’s orientation, respectively. Fig. 8 plots the tracking of the contact forces of the left-front foot. The feedback ratio $\int |\tau_{PD}|/|\tau|$ is a good metric to determine how accurate our kinematic/dynamic model (e.g. body inertia and estimation of the CoM) of the robot is. In particular the feedback ratio represents the contribution of the PD controller relative to the total commanded torque. The feedback ratio

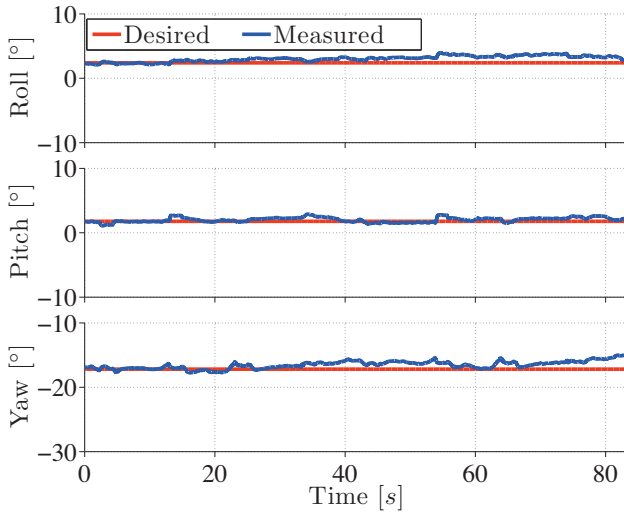


Fig. 7: Experimental results. Tracking of the base’s orientation during the walking.

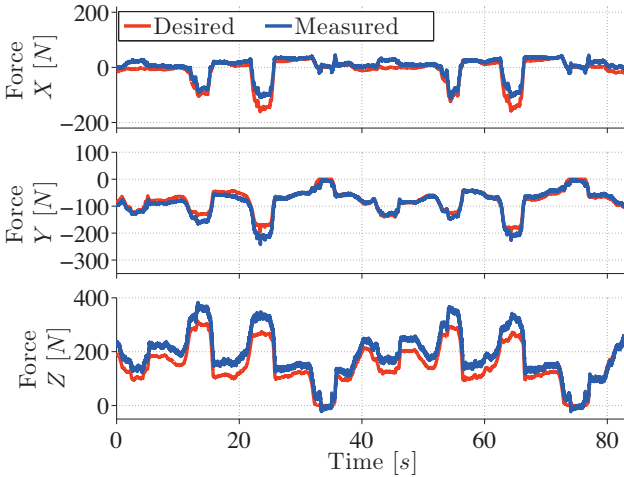


Fig. 8: Cartesian components of the contact forces in the left-front leg. Red plots are the desired forces generated by the optimizer, while blue plots are the actual contact forces. The difference is due to the action of the PD controller, whose overall influence is on average lower than 18%.

computed for the experimental data of Fig. 8 is 18%, which shows a very small intervention of the PD feedback action during the test. Fig. 9 shows the distribution of the GRFs on all the legs for the same groove experiments. The GRFs are always inside the friction-pyramid boundaries. Note that the unilateral constraints on the contact forces implicitly restrict the CoP inside the convex hull of the support polygon.

4.2.3 Torque limits

During the walk the robot reached configurations in which the torques needed at the *HFE* joints exceeds their limits.

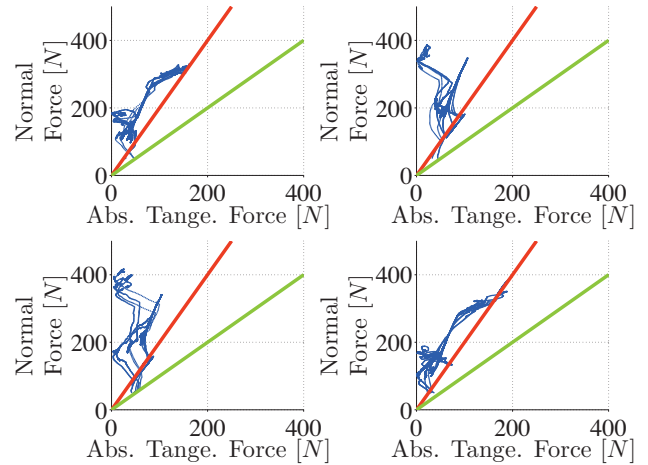


Fig. 9: Distribution of the contact forces at the four feet. The plots show the forces along the ground normal direction as a function of the norm of the tangential forces. The green lines represent the estimated boundaries of the friction cones, which correspond to a friction coefficient $\mu = 1$, while the red line represent the conservative friction coefficient of $\mu = 0.5$ set in the controller.

Indeed for the sagittal joints the available torque depends on the joints’ positions because the lever arm of the piston varies (nonlinearly) with the joint’s angle [22]. We therefore tuned the matrix W_τ to penalize torques at the *HFE* joints. We also tried to repeat the experiment with a steeper wall inclination $\theta = 60^\circ$, both in simulation and on the robot. The experiment failed because both *HFE* and *HAA* reached their torque limits and the problem could not be solved by tuning W_τ (see Section 5.3). Conversely in simulation, where the torque limitations were absent, the test succeeded.

4.2.4 Comparison with other approaches

We implemented three other algorithms to compare them with our approach on the same experimental conditions:

1. a high-gain joint PD position controller (with $K = 500 \text{ Nm/rad}$ and $D = 6 \text{ Nms/rad}$);
2. our controller, but without considering the terrain inclination (i.e. $\theta = 0^\circ$);
3. a low-gain PD controller ($K = 150 \text{ Nm/rad}$ and $D = 3 \text{ Nms/rad}$) superimposed to a floating-base gravity compensation [19].

We computed the floating-base gravity compensation as:

$$\tau_{ff} = (N_c S^T)^\sharp N_c g, \quad (15)$$

where $N_c = I - J_c^\sharp J_c$ is the null-space projector of the contact Jacobian $J_c \in \mathbb{R}^{k \times (n+6)}$, which is a stack of the stance feet’s Jacobians $J_{c_i} = [J_{B_i} J_{q_i}]$, $(\cdot)^\sharp$ is the Moore-Penrose

pseudoinverse, and $g \in \mathbb{R}^{n+6}$ are the generalized forces due to gravity. With all these controllers the robot has lost the traction with the surface when moving the body, demonstrating the importance, for this kind of task, of controlling the GRFs. The first controller does not have an optimization stage and so the feet quickly start to slip. The second controller directs the GRFs on the vertical axis (Z), so once the GRFs leave their friction cones the robot slips and falls. The last controller compensates for gravity using a Moore-Penrose pseudoinverse, which generates a minimum-norm torque vector. This generally corresponds to GRFs pointing through the hip-joint axis. Even though the GRFs could possibly lie inside the friction cones, the lack of an explicit optimization results in the robot slipping and falling when the robot's trunk starts moving.

5 Practical Issues

Here we present a number of steps taken to ensure the robustness of the robot's behavior in a real-world environment.

5.1 Friction Cone Estimation

Before performing the walking experiments we estimated the friction coefficient μ at the contact between the rubber coating of the robot's feet and the wall surface. We laid one of the groove walls flat on the ground, with the robot standing statically on top of it. Then we made the robot exert horizontal GRFs, increasing up to the point at which one of the feet slipped. Finally, we chose $\mu = \sqrt{f_x^2 + f_y^2}/f_z$, where f_x, f_y and f_z are respectively the two tangential components and the normal component of the contact force at the foot, right before slipping.

5.2 Identification of the CoM with Static Poses

In order to improve the estimation of the center of mass of the robot we identified its location. Since most of the mass of the robot is located in the trunk, we assumed that the CoM does not depend on the configuration of the legs – as we did in the controller design. This allows us to consider just a lower dimensional model of the robot (e.g. the rigid body of the trunk). When the robot is still (i.e. $\dot{q} = \ddot{q} = 0$) the net moment at the CoM is zero:

$$\Gamma_{com} = \sum_{i=0}^3 (f_i \times p_{com,i}) = \sum_{i=0}^3 f_i \times (p_i - x_{com}) = 0, \quad (16)$$

where $f_i \in \mathbb{R}^3$ is the GRF at the i^{th} foot and $p_{com,i} \in \mathbb{R}^3$ is the distance from the CoM to the i^{th} foot. The only unknown in this equation is the CoM position x_{com} . By collecting force

and position measurements over T seconds while the robot was in a set of manually designed static poses, we could write the overconstrained system of equations:

$$\underbrace{\begin{bmatrix} [\sum_{i=0}^3 f_i(0)]_{\times} \\ \vdots \\ [\sum_{i=0}^3 f_i(T)]_{\times} \end{bmatrix}}_A x_{com} = \underbrace{\begin{bmatrix} \sum_{i=0}^3 (f_i(0) \times p_i(0)) \\ \vdots \\ \sum_{i=0}^3 (f_i(T) \times p_i(T)) \end{bmatrix}}_b \quad (17)$$

We designed the static poses to obtain a sufficiently rich regression matrix A . We then estimated the CoM's position as $\hat{x}_{com} = (A^T A)^{-1} A^T b$. The estimated CoM lied at about 2.7cm (in the xy plane) from the CoM computed from the CAD model. Moreover, by performing a recursive least-squares estimation with *forgetting factor*, we measured how much the CoM's estimation varied through all the static poses due to the influence of the mass of the legs. The variations were of $\approx 1\text{cm}$; this suggested that approximating the robot's CoM with the trunk's CoM was acceptable for our quasi-static movements.

5.3 Torque Minimization

The joint torque limits proved to be a crucial issue during our experiments. The respect of the joint-torque limits can be achieved in (8) through either the cost function or the inequality constraints. Even though this allows constraint violations, we used the first method because the second one was computationally too expensive. The regularization term $f^T W f$ can be defined in order to penalize joint torques rather than GRFs. This can be achieved by knowing the relationship between feet forces and torques: $\tau = -S J_c^T f$. Therefore to minimize $\tau^T W_\tau \tau$, with $W_\tau \in \mathbb{R}^{3c \times 3c}$ being a diagonal positive-definite matrix, we set

$$W = J_c S^T W_\tau S J_c^T$$

This results in implicitly minimizing the torques of the stance-legs' joints.

5.4 Robustness to Friction Coefficient

Looking at Fig. 9 it can be noted that GRFs are always close to the cone boundaries. This is expected because, due to the quasi-static motions, gravitational components (mainly vertical) dominates in the body wrench, and using a regularization that minimizes the norm of the torques or of the forces leads to solutions that are close to the cone boundaries (for the actual task). To improve robustness it would be preferable to have a solution where forces are close to the cones' normals. This is equivalent to penalizing the norms of the feet's forces in frames that are aligned with the contacts'

normals. To achieve this we could set the following block-diagonal weight matrix [20]:

$$W = \begin{bmatrix} T_0 W_{n_0} T_0^T & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & T_c W_{n_c} T_c^T \end{bmatrix}, \quad (18)$$

where $T_i = [t_{1_i} \ t_{2_i} \ n_i]$ is a rotation matrix whose columns are the coordinate axis of a frame aligned with the contact surface i . The weight matrix for each stance leg i is $W_{n_i} = \text{diag}(K_{t_1}, K_{t_2}, 1)$, where K_{t_1} and K_{t_2} are the weights used to penalize the tangential forces in the t_{1_i} and t_{2_i} directions. Despite this regularization would be preferable for the robustness of the controller, due to the torques' limitations we used the regularization described in Section 5.3 in the real experiments.

6 Conclusions and future work

We presented a self-contained planning/control framework for quadrupedal quasi-static walking on high-sloped terrain, reporting experimental results on a torque-controlled quadruped robot. By direct control of the GRFs we could avoid slippage despite the high terrain inclination (i.e. 50°). Similar theoretical control architectures have been presented in recent years [10, 11, 6, 14], but to the best of our knowledge, the few demonstrations on torque-controlled platforms have been limited to humanoid balancing [9, 23, 17] and quadruped locomotion on terrains with low slope ($\leq 40^\circ$) [20, 8]. The presented experiments show that the recent trend of force-based control frameworks can be used to perform locomotion on high-slope terrain. We believe that this capability is essential for the deployment of robots in adverse environments, such as mountains or disaster-recovery scenarios.

In the controller we assumed that the CoM does not depend on the configuration of the legs, though their mass is far from negligible. Despite this simplifying assumption, the use of a lower-dimensional model was sufficient to perform the task. Furthermore, we have shown that a simple procedure is adequate to estimate the few inertial parameters used in our simplified model.

In the near future we plan to relax the simplifying assumptions undertaken in this work (quasi-staticity, lower-dimensional model) and develop a whole-body control framework with optimization of GRFs, joint torques and joint limits. This framework will be suitable to perform more *dynamic* tasks. Indeed, relaxing the quasi-static assumption (i.e. computing the whole-body dynamics) would allow for more aggressive movements, hence faster locomotion. We want to speed up the controller in order to solve the optimization in real-time, despite the increased computational burden (due to more inequalities and variables).

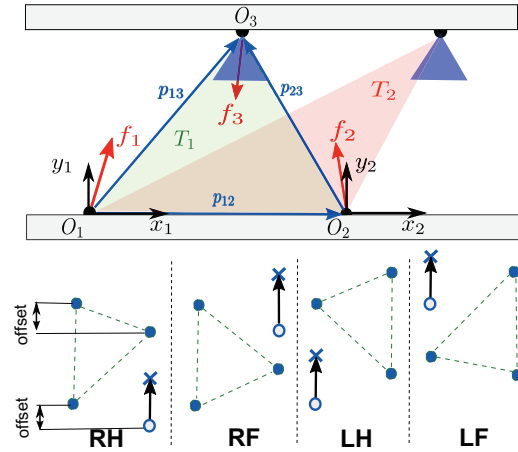


Fig. 10: (top) Top view of two different support triangles T_1 and T_2 . Relative to T_1 we report also the three contact forces f_1, f_2, f_3 , the distance between the contact points p_{12}, p_{13}, p_{23} and the friction cone of f_3 . (bottom) Gait sequence for the groove walk experiments.

We plan to perform more challenging tasks like locomotion on different groove shapes (e.g. diverging walls, irregular slopes, turns), on ice and slippery slopes (low friction) and on a moving platform (keep balance).

The framework will also be extended to our Centaur robot (a quadruped base with two arms on top) in order to perform whole-body manipulation tasks. In this scenario the legs can provide assistance to pull or push an object. Furthermore, a body-posture optimization will be implemented with the purpose to increase stability and be more effective to exert a force in a desired direction, while minimizing the torques at the legs's joints. This is a strategy, to reduce the overall energy expenditure, which is very common for humans.

More advanced techniques for the estimation of the base's position/orientation [1] could improve the performances of the controller. Finally, we plan to incorporate more information on the geometry of the environment, possibly combining vision and active *haptic* exploration (e.g. touching three points on the terrain and fitting a plane).

A Intuitive justification of foot placement

This section explains our choices regarding foot positioning for quadrupedal walking on v-shaped terrain. We show that, when the robot stands on three feet, having an *acute* support triangle is convenient for maintaining the robot in equilibrium. We know that the robot is in equilibrium when the net external force and moment (about any point) acting on it are zero. We define a reference frame O_1 located at foot 1 (see Fig. 10), with the axis z_1 aligned with gravity and the axis x_1 pointing towards foot 2 (which we assume to be approximately aligned with foot 1). At the equilibrium, the net moment $m \in \mathbb{R}^3$ about z_1 has

to be zero, that is:

$$P_z m = (P_{xy} p_{12}) \times (P_{xy} f_2) + (P_{xy} p_{13}) \times (P_{xy} f_3) = 0, \quad (19)$$

where $P_{xy} \in \mathbb{R}^{3 \times 3}$ projects onto the $x_1 y_1$ plane, $P_z \in \mathbb{R}^{3 \times 3}$ projects onto the z_1 axis, $f_2, f_3 \in \mathbb{R}^3$ is the GRF at the foot 2 (3), and $p_{12}, p_{13} \in \mathbb{R}^3$ are the lever arms from foot 1 to foot 2 and 3, respectively. The first term of (19) always generates a positive moment about z_1 because of the unilaterality constraints, i.e. $f_{2y} > 0$. To have equilibrium then we need f_3 (i.e. the second term) to generate a negative moment about z_1 . In other words $(P_{xy} f_3)$ must lie in the right halfspace delimited by the line passing through feet 1 and 3. Similarly, computing the net moment about z_2 (i.e. the z axis of the frame O_2), we can infer that to have equilibrium $(P_{xy} f_3)$ must lie in the left halfspace delimited by the line passing through feet 2 and 3. This implies that $(P_{xy} f_3)$ must lie — not only inside the friction cone, but also — inside the *support cone*, that is the cone originating in O_3 and delimited by two sides of the support triangle (green cone in Fig. 10). We can then state that having an *acute* support triangle leaves more freedom in the choice of f_3 because it results in a bigger area of intersection between the friction cone and the support cone. If p_3 gets too close to p_1 or p_2 , a part of the friction cone of f_3 stops intersecting the support cone, leaving less freedom for the choice of f_3 (e.g. red support triangle in Fig. 10).

Taking advantage of these insights we planned contact configurations that generate acute support triangles. A gait sequence that satisfies this requirement is *RH, RF, LH, LF*, in which we set an initial offset positions for the feet along the x direction (see Fig.10 (bottom)).

Acknowledgments

This research has been funded by the Fondazione Istituto Italiano di Tecnologia.

References

1. Bloesch M, Hutter M, Hoepflinger M, Leutenegger S, Gehring C, Remy CD, Siegwart R (2012) State Estimation for Legged Robots-Consistent Fusion of Leg Kinematics and IMU. *Robotics: Science and Systems*
2. Boaventura T, Semini C, Buchli J, Frigerio M, Focchi M, Caldwell DG (2012) Dynamic torque control of a hydraulic quadruped robot. 2012 IEEE International Conference on Robotics and Automation pp 1889–1894
3. Buchli J, Kalakrishnan M, Pastor P, Schaal S (2009) Compliant quadruped locomotion over rough terrain. *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009 IROS 2009*
4. Cheng G, Hyon Sh, Ude A, Morimoto J, Hale JG, Hart J, Nakanishi J, Bentivegna D, Hodgins J, Atkeson C, Mistry M, Schaal S (2008) CB: Exploring neuroscience with a humanoid research platform. 2008 IEEE International Conference on Robotics and Automation pp 1772–1773
5. Frigerio M, Buchli J, Caldwell DG (2012) Code generation of algebraic quantities for robot controllers. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems pp 2346–2351
6. Gehring C, Coros S, Hutter M, Bloesch M, Hoepflinger M, Siegwart R (2013) Control of Dynamic Gaits for a Quadrupedal Robot. *IEEE International Conference on Robotics and Automation (ICRA)*
7. Gertz E, Wright S (2001) OOQP user guide. Math and Comp Sci Division Techn Memorandum
8. Hutter M, Hoepflinger M, Gehring C (2012) Hybrid Operational Space Control for Compliant Legged Systems. In: *Proceedings of Robotics: Science and Systems, 2012.*
9. Hyon S, Hale J, Cheng G (2007) Full-body compliant human-humanoid interaction: balancing in the presence of unknown external forces. *Robotics, IEEE Transactions on* 23(5):884–898
10. de Lasa M, Mordatch I, Hertzmann A (2010) Feature-based locomotion controllers. *ACM Transactions on Graphics* 29(4):1
11. Lee S, Goswami A (2010) Ground reaction force control at each foot: A momentum-based humanoid balance controller for non-level and non-stationary ground. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2010* pp 3157–3162
12. Lee SH, Goswami A (2012) A momentum-based balance controller for humanoid robots on non-level and non-stationary ground. *Autonomous Robots* 33(4):399–414
13. Lin PC, Komsuoglu H, Koditschek D (2005) A leg configuration measurement system for full-body pose estimates in a hexapod robot. *IEEE Transactions on Robotics* 21(3):411–422
14. Macchietto A, Shelton CR (2009) Momentum Control for Balance. In: *ACM Transactions on Graphics (TOG)*
15. Mistry M, Schaal S, Yamane K (2009) Inertial parameter estimation of floating base humanoid systems using partial force sensing. In: *Humanoids 2009, IEEE International Conference on*
16. Orin DE, Goswami A, Lee SH (2013) Centroidal dynamics of a humanoid robot. *Autonomous Robots* 35(2-3):161–176
17. Ott C, Roa Ma, Hirzinger G (2011) Posture and balance control for biped robots based on contact force optimization. 2011 11th IEEE-RAS International Conference on Humanoid Robots pp 26–33
18. Pratt J, Chew C, Torres A, Dilworth P, Pratt G (2001) Virtual model control: An intuitive approach for bipedal locomotion. *The International Journal of Robotics Research* 20:129–143
19. Righetti L, Buchli J, Mistry M, Schaal S (2011) Inverse dynamics control of floating-base robots with external constraints: A unified view. 2011 IEEE International Conference on Robotics and Automation pp 1085–1090
20. Righetti L, Buchli J, Mistry M, Kalakrishnan M, Schaal S (2013) Optimal distribution of contact forces with inverse dynamics control. *The International Journal of Robotics Research* (January)
21. Schaal S (2001) The S L Simulation and Real-Time Control Software Package
22. Semini C, Tsagarakis NG, Guglielmino E, Focchi M, Cannella F, Caldwell DG (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
23. Stephens BJ, Atkeson CG (2010) Dynamic Balance Force Control for compliant humanoid robots. 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems pp 1248–1255

Link to the video

[Click here to download Electronic Supplementary Material \(Video, Movie, Audio, etc.\): video_link.txt](#)

Michele Focchi is currently a post-doc researcher at the Advanced Robotics department of Istituto Italiano di Tecnologia. He holds a Master of Science in Control System Engineering from Politecnico di Milano in 2007. In 2009 he joined the Advanced Robotic Department (ADVR) at Istituto Italiano di Tecnologia developing a prototype of Novel concept for an air-pressure driven micro-turbine for power generation for which he obtained an international patent and several awards. In 2010 he started a PhD getting involved in the Hydraulically Actuated Quadruped Robot project at the Dynamic Legged Systems Laboratory at ADVR. His research interests range from dynamic locomotion with legged robots, to actuation technologies. He was also involved in the design of components for the HyQ quadruped robot and in the implementation of low-level controllers for locomotion purposes.

Andrea Del Prete was born in Cesena (Italy) in 1984. He received his degree in Computer Engineering (with honors) from the 2nd faculty of the University of Bologna (Italy) in 2009. In 2013 he took his PhD from the Italian Institute of Technology in Genoa, with a thesis titled "Control of Contact Forces using Whole-Body Force and Tactile Sensors: Theory and Implementation on the iCub Humanoid Robot". Since 2014 he has been a PostDoc at LAAS-CNRS in Toulouse (France), where he is currently working on whole-body motion generation for the humanoid robot HRP-2.

Ioannis Havoutis is a post-doctoral researcher at the Department of Advanced Robotics in the Italian Institute of Technology. His research combines whole-body motion planning and control with machine learning, focusing on robots with legs and arms. Previously he received his PhD in Robotics and his M.Sc. in Artificial Intelligence with Distinction from the School of Informatics at University of Edinburgh and holds a Ptychion (B.Sc.) in Computer Science from the Department of Informatics at the Technological Educational Institute of Athens. His research interests include legged locomotion, dynamic motion and path planning, force and torque control, machine learning and optimal control.

Dr. Featherstone obtained his Bachelor's degree in Mathematics with Electronics from Southampton University in 1979, and his Ph.D. in Artificial Intelligence from Edinburgh University in 1984. He spent more than 6 years working in industry, first for a small start-up company in the UK, and then for Philips Electronics in the USA. In 1992 he moved to Oxford University to take up an EPSRC Advanced Research Fellowship. From 1998 to 2001 he was a Lecturer in Computer Science at the University of Wales, Aberystwyth, and from 2001 to 2011 he worked for the Australian National University, holding positions in both Computer Science and Engineering. He is currently a Visiting Professor at the Italian Institute of Technology. He is also the inventor of the articulated-body algorithm, the author of two books on robot and rigid body dynamics, and a Fellow of the IEEE.

Claudio Semini is the head of the Dynamic Legged Systems lab of the Department of Advanced Robotics at Istituto Italiano di Tecnologia (IIT). He holds an MSc degree from ETH Zurich in electrical engineering and information technology (2005). From 2004 to 2006 he first visited the Hirose Lab at the Tokyo Institute of Technology, followed by work on mobile service robotics at the Toshiba R&D center in Kawasaki, Japan. During his doctorate from 2007-2010 at the IIT he designed and constructed

the quadruped robot HyQ and worked on its control. After a post-doc in the same department, in 2012, he became the head of the Dynamic Legged Systems lab. His research focus lies on the construction and control of highly dynamic and versatile legged robots in real-world environments.

Darwin G Caldwell is a founding Director at the Italian Institute of Technology in Genoa, Italy, and a Honorary Professor at the Universities of Sheffield, Manchester, Bangor, Kings College, London and Tianjin University China. His research interests include innovative actuators, humanoid and quadrupedal robotics and locomotion (iCub, cCub, HyQ and COMAN), haptic feedback, force augmentation exoskeletons, dexterous manipulators, biomimetic systems, rehabilitation and surgical robotics, telepresence and teleoperation procedures. He is the author or co-author of over 450 academic papers, and 17 patents and has received awards and nominations from several international journals and conference including; IMechE Best Paper Award 2009, Ind. Robot Journal 2010, ICRA (2007), IROS (2007, 2012, 2013), ICAR (2003), Humanoids (2008, 2012), CASE (2008), ICMA (2011), Robio (2013) IFAC IAV, MMVR (2011), ACHI (2010), WorldHaptics (2007) and Virtual Concepts (2006). He is secretary of the IEEE/ASME Trans. On Mechatronics and on the editorial board of the International Journal of Social Robotics and Industrial Robot.

Author Photographs
[Click here to download high resolution image](#)







Author Photographs
[Click here to download high resolution image](#)





