Viscosity-based Height Reflex for Workspace Augmentation for Quadrupedal Locomotion on Rough Terrain

Michele Focchi, Roy Featherstone, Romeo Orsolino, Darwin G. Caldwell, and Claudio Semini
Department of Advanced Robotics, Istituto Italiano di Tecnologia, (IIT), Genova, Italy
{*michele.focchi, roy.featherstone, romeo.orsolino, darwin.caldwell, claudio.semini*}*@iit.it*

***Abstract*— We propose a reactive locomotion strategy, called *height reflex*, that is useful to address big elevation changes in the terrain (e.g. when a quadruped robot has to step down from a high platform). In these cases the swing leg can lose mobility creating issues in the subsequent steps. The *height reflex* is a foot trajectory replanning strategy that redistributes the swing motion (in a smart way) to the stance legs to "lower" the whole trunk and to aid the foothold searching motion. To spread the motion we exploit a massless link model of the robot with *virtual* dampers at the joints, which is used to replan the feet trajectories. The proposed approach is able to incorporate kinematic limits, it is easy-to-tune, computationally efficient and suitable for real-time implementations. The reflex is implemented and experimentally evaluated on the 80 kg hydraulic quadruped HyQ. With our approach we were able to address high steps, up to 24 cm which is 30% of HyQ leg length and 53% of its retractable leg range.**

I. Introduction

Legged robots are are often designed to traverse unstructured environments that, in some cases, can pose significant challenges when high steps (level change) are present.

These terrain features can be successfully addressed by *whole body trajectory optimization* when a map of the environment is available. Despite recent advances in this field [1], [2], [3], optimal planning is still far from being realized in real time, due to the complexity of the optimization process and the need to have both reasonably accurate 3D maps of the environment [4] and reliable state estimation [5], [6], [7] that relies on accurate IMU or proprioceptive sensing.

On the other hand, these approaches are of limited applicability in cases of visual *deprivation* where a *reactive strategy* is always preferable [8], [9]. Indeed, in nature reflexes are dominant for various kinds of adjustments in medium-speed walking of animals [10]. This motivated us to have a *blind* locomotion layer which quickly reacts to unpredictable situations such as rolling stones, big slippages, external pushes and abrupt terrain changes. Earlier implementations of reflexes on robots date back to 2001 with the BISAM robot project [11]. In this context reflexes substantially increased the robustness of locomotion. In particular, a reflex-based posture control was implemented via fuzzy control and reinforced learning by correcting CoG and posture trajectories based on foot force sensor input. The main motivation was to address a broad range of obstacles without the need to model them.

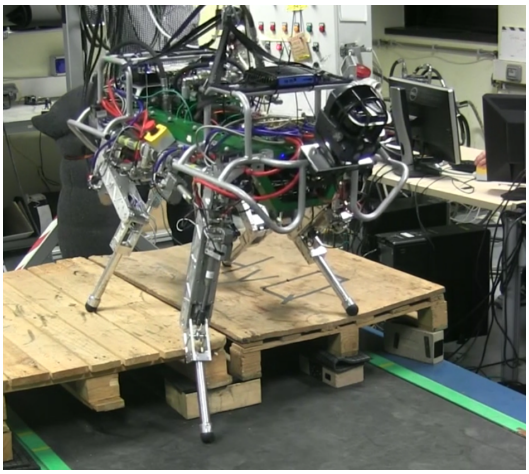

Fig. 1: HyQ robot stepping down from a 24 *cm* high platform.

In this work we propose a *height reflex* strategy to successfully negotiate big height changes (e.g. when the robot steps down from a high pallet).

In this context, an important issue is the *loss of mobility* of the swing leg, when trying to establish a new foothold during a *searching* motion [12]. Indeed, in our crawl locomotion strategy [12], a haptic touchdown is implemented, which is crucial for rough terrain locomotion. Haptic touchdown means that a "*searching motion*" is performed with the swing leg, that terminates only when a stable foothold is established (or workspace limits are reached). [1] In the case of our quadruped HyQ [13], a loss of mobility can occur when the leg gets close to its kinematic singularity (e.g. the leg stretches completely). This is the case when the robot is stepping down from a high step. [2] An elegant way to address this situation, is to "lower" the whole trunk to aid the searching motion. In other words, the swing motion can be "reconfigured" by mapping it among the four legs. This realizes the searching motion without stretching too much the swing leg. Another way of seeing this is that the height reflex is able to extend the workspace of the swing leg, through the motion of the body. The underlying idea, is that, for a legged robot which is statically walking, the same

[1] To assess the touchdown we check if the value of the contact force (estimated by torque sensing) project along the searching direction, goes beyond a certain minimum threshold (e.g. 30N).

[2] See accomp. video youtu.be/FPkvu29WLUc

motion of the foot (in an inertial frame) can be achieved just by moving the swing leg joints (e.g. to move the foot), or with a combination of all the legs (trunk motion + swing motion). This exploits a motion in the null-space of the stance constraints, inherent in the kinematic redundancy of a quadruped robot. This behaviour is somewhat similar to the one presented in [14], enforcing specific constraints (in a hierarchical fashion) in a whole body optimization approach for their robot Anymal. In that paper Anymal proved to be able to step down from a 14 *cm* high step (23% of the maximum leg length). Specifically, the authors set a constraint on the vertical body acceleration which changed the robot height according to manipulability measures of all legs. In essence, the originally planned trajectories were modified by the whole body control to fulfil this constraint. However, doing this at the control level, means that the actual trajectory will necessarily drift away from the planned one when the reflex is active, creating a tracking error (e.g. 0.2 *rad* in the pitch). The authors do not give details about it, but it seems that the responsibility to recover this error in the following locomotion phases, is given to the controller. This can become an issue when facing big terrain changes. Instead of having the controller trying to enforce the desired behaviour, we decided to tackle the problem from a *planning* perspective. The main contribution of this work is a lightweight, easy-to-tune reflex strategy (*height reflex*) that modifies the original plan, by *online* replanning the feet trajectory (end therefore the Center of Mass (CoM) trajectory) in order to encode a desired "squatting" behaviour. We demonstrated the effectiveness of the proposed algorithm showing our 80 kg quadruped robot HyQ, negotiating step elevations *higher* than the state of the art [15], [16], [14]. [3] We showed HyQ stepping down from a 24 *cm* high platform which is 30% of the leg length and 53% of the retractable leg range [13]. The proposed approach is also able to incorporate kinematic limits.

The paper is structured as follows: in Section II we briefly illustrate our statically stable crawl locomotion framework [12] while in Section III we describe the actual implementation of the *height reflex* strategy. Section III-E shows how the parameters in the *damping model* can be adjusted to achieve the desired height reflex behaviour with a brief introduction on how to evaluate the leg mobility (Section III-F). In Section IV we present experimental results. Finally, in Section V we address the conclusions.

II. Crawl Locomotion Framework

In this section we briefly introduce our statically stable crawl framework that we use for rough terrain locomotion and show how the *height reflex* is incorporated in it. Figure 2 shows a block diagram of the framework. The core module is a state machine (see [12] for details) that, as an orchestra leader, switches between two temporized/event-driven locomotion phases: a swing phase, and a body motion phase. During the *body motion phase* the robot CoM is shifted onto the future support triangle, which is opposite to the next swing leg, in accordance to a user-defined foot sequence. The *swing phase* has a *haptic* touchdown with a *searching* motion. According to this, the swing motion does not stop in a pre-scheduled way, but the leg keeps extending until a new touchdown is established [12]. This occurs when the *haptic contact handler* module detects a contact through thresholding on the Ground Reaction Forces (GRFs) or directly via a foot contact switch. A *searching* motion with *haptic* touchdown is crucial for terrain adaptation which is a fundamental feature when addressing locomotion across rough terrain. This is important also when vision is used (there are always tracking errors and the accuracy of a height map used for locomotion is typically in the order of *cm* [1], [2]). The step length L_i, is computed from user-desired linear/angular velocity (v_f, v_h). Both the body and the swing trajectories are generated as quintic polynomials in a frame which is aligned to the terrain (*terrain frame*). We map the body motion into feet motion to provide, after inverse kinematics, a joint reference for our impedance (PD) controller which is implemented at the joint level. The *whole body control* module [12] is useful to improve the tracking of the CoM and redistribute the weight among the stance legs (c_{st}) ensuring that friction constraints (μ) are not violated. Its output is a feed-forward action τ^d_{ff} that is added to form the torque reference τ^d that is sent to the low level joint torque controller.

When locomoting through rough terrain, the shape and inclination of the support polygon can significantly change from one step to the other. In particular, the orientation of the robot should be adapted to match the terrain shape to avoid hitting the kinematic limits. In addition to this, the new CoM X^{tg}_{com} and body orientation R^{tg} targets are computed at each step, considering the *actual* robot state. This prevents error accumulation e.g. due to slips, stones that can roll away under the feet. [4] These *new* terrain adaptation features were not presented in [12], and are incorporated in the state machine block (Fig. 2).

The height reflex module is active only during the swing phase when the base is not moving. As mentioned in the introduction, when the swing leg is extended too much it can can end up in an inconvenient kinematic configuration (usually associated with limited mobility). Most of the time, this creates issues in the subsequent step of the robot, since it is in a stretched leg configuration. With a *height reflex* we want to mitigate this problem in such a way that, when the swing leg extends too much, losing mobility, the other 3 stance legs retract aiding the foothold searching motion.

In addition to this, in the case that a kinematic limit of a joint is approached, the reflex corrects the feet trajectories such that the joint desired positions are kept at their usable range. When the reflex is active, the corrected feet trajectories

[3] In the Darpa Robotics Challenge's car egress task, the 180 *cm* tall Atlas robot was shown to step down from a 20 *cm* high platform (see a video of team MIT's ATLAS performing the egress task).

[4] A video showing the importance of terrain adaptation in walking experiments on very rough terrain is available at https://youtu.be/pQPO5exJQd0, height reflex activation is at 00:54.

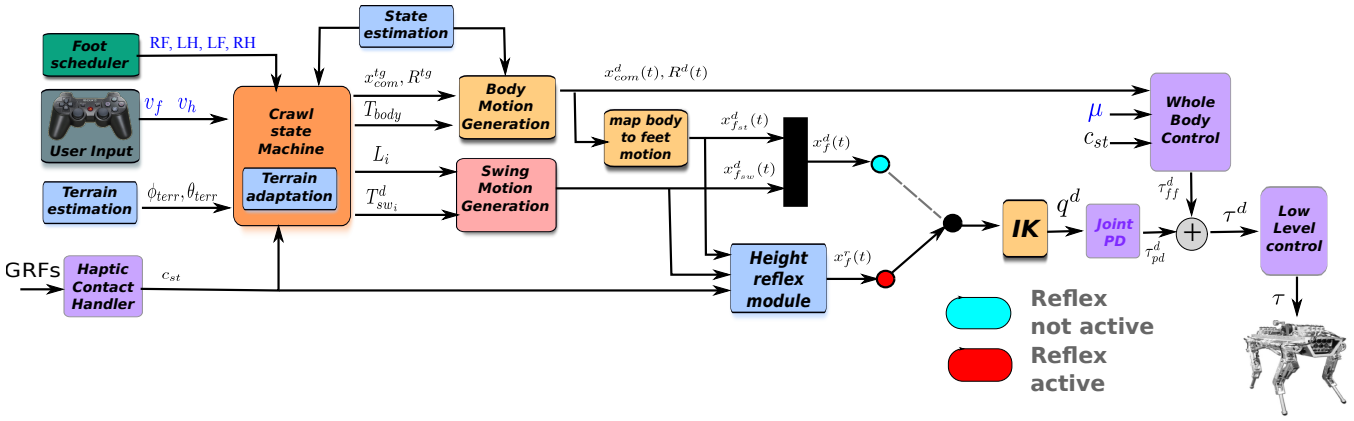

Fig. 2: Block diagram of the locomotion framework, the *height reflex* module is depicted in light blue.

x^r_f are sent to the inverse kinematics module instead of the original plan.

III. Height Reflex Algorithm

In this section we describe the implementation of the *height reflex* and how the original (x^d_f) feet trajectories are modified (x^r_f) when the reflex is active, spreading the motion among the 4 legs when the swing leg is losing mobility or stopping it when some joint is reaching the kinematic limits. Figure 3 pictures a detailed scheme of the height reflex module presented in Fig. 2. Henceforth, we will consider a partition of feet variables $\# = \begin{bmatrix} \#_{sw} & \#_{st} \end{bmatrix}$ into the swing part $\#_{sw} \in \mathbb{R}^3$ and the stance part $\#_{st} \in \mathbb{R}^{n-3}$. Where # can be either a *virtual* force $f^r \in \mathbb{R}^n$, the original $x^d_f \in \mathbb{R}^n$, or the reflex-corrected $x^r_f \in \mathbb{R}^n$ feet trajectories or Jacobians, respectively. For the forthcoming sections, all reflex-related variables will be denoted with the superscript r. Since our robot HyQ [13] has 12 degrees of freedom (DoFs) we will consider $n = 12$. Note that a *virtual* force is not a real force but just a variable which is convenient in the online replanning strategy.

A. Virtual Damper Model

In this section we introduce a massless model of the robot with (variable) viscosity (damping) at the joints (thus a n-order system, where $n = 12$) that we will use to be able to "redirect" the motion from the swing leg to the stance legs (see Fig. 4). By setting different viscosities we can "redirect" the motion toward the joints/legs with lower viscosity (see damping schedule Section III-E). A feature of this simplified model is that we can easily regulate the *transfer of motion* by simply tuning the virtual damping value for each joint. The height reflex can be seen as setting a "honey-like" environment for the leg which is losing mobility while reducing "viscosity" for the other legs that are in stance that will start to move instead. Alternatively, scheduling the damping in this way is somewhat similar to a continuously variable transmission (CVT) where the motion is "redirected" to where it is more convenient. A damper model (or viscosity model) relates joint torques $\tau^r \in \mathbb{R}^{12}$ to joint velocities $\dot{q}^r \in \mathbb{R}^{12}$ according to the known relationship:

$$\tau^r = D_r \dot{q}^r \tag{1}$$

where $D_r \in \mathbb{R}^{12\times12}$ is the virtual damping matrix of all the active joints.

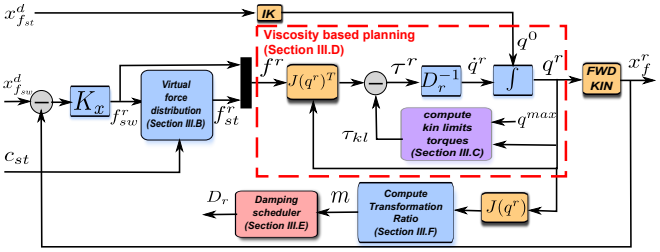

Fig. 3: Block diagram of the height reflex module. Inputs are the originally planned desired feet position (swing $x^d_{f_{sw}} \in \mathbb{R}^3$ and stance $x^d_{f_{st}} \in \mathbb{R}^9$) and the set c_{st} of stance legs while the output is the corrected feet positions x^r_f.

B. Virtual Force Distribution

The reflex-replanned trajectory is the resulting motion of the damper model in response to virtual (linear) forces f^r at the feet (see Fig. 3). The first step is to translate the desired swing foot motion $x^d_{f_{sw}} \in \mathbb{R}^3$ to the force level (e.g. a Cartesian force $f^r_{sw} \in \mathbb{R}^3$) by means of a virtual spring (e.g. impedance K_x, see Fig. 5).

$$f^r_{sw} = K_x(x^d_f - x^r_f) \tag{2}$$

where $K_x \in \mathbb{R}^{3\times3}$ is the Cartesian stiffness matrix (an upper bound is set to avoid numeric instability). Then we (statically) map f^r_{sw} to the stance feet $f^r_{st} \in \mathbb{R}^9$ in a least-square fashion. [5] In our problem we have 9 unknowns (f^r_{st}): 3 stance legs with point feet (3 components each). We enforce 6 constraints for static equilibrium. Namely, 3 constraints for the equilibrium of linear forces $f^r_i \in \mathbb{R}^3$ and 3 constraints for

[5] In theory we could have done the mapping at the hip points instead of the stance feet. However, in that case we would have to consider the whole wrench rather than just a pure force, adding unneeded complexity to the problem.

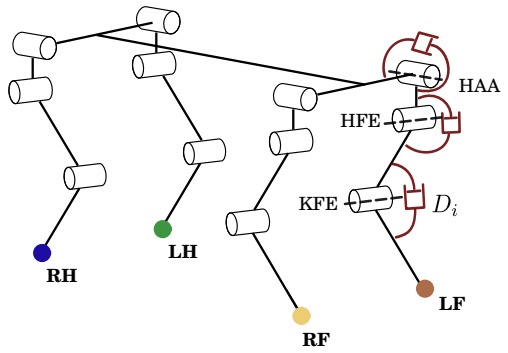

Fig. 4: Simplified kinematic model of the robot with viscosity at the joints and massless links. The four legs are called Left Front (LF), Right Front (RF), Left Hind (LH), Right Hind (RH). The three leg joints are called Hip Abduction/Adduction (HAA), Hip Flexion/Extension (HFE) and Knee Flexion/Extension (KFE).

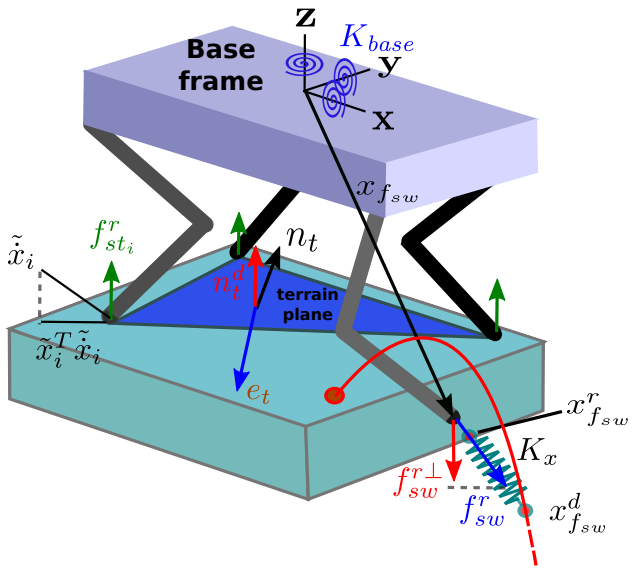

Fig. 5: Schematic of the vectors and frame definitions for the height reflex algorithm.

the equilibrium of moments (about the swinging point foot) $M_i^r \in \mathbb{R}^3$:

$$\begin{aligned} \sum_{i \in c_{st}} f_i^r &= -f_{sw}^{r\perp} \\ \sum_{i \in c_{st}} M_i^r &= \sum_{i \in c_{st}} (x_{f_i}^r - x_{f_{sw}}^r) \times f_i^r = m \end{aligned} \tag{3}$$

where c_{st} is the set of the stance legs, and f_i^r is the virtual force and $x_{f_i}^r$ is the corrected foot position for the $i-th$ foot, respectively. Unless differently specified, all vectors are expressed in the base frame. To ensure that static stability is preserved during the reflex motion we need to constrain this motion to be only vertical (e.g. only the component of f_{sw}^r along gravity is mapped: $f_{sw}^{r\perp} = e_z e_z^T f_{sw}^r$). This mapping should also result in motions of the stance legs that are kinematically consistent (e.g. no internal forces should be created between the stance legs, and the support feet do not move). We can enforce the velocity of the three supporting feet to be zero (in the world frame), however, this would mean to make these constraints dependent on an estimation of the robot state that can possibly drift or be affected by errors. Therefore, we rephrase these constraints in the base frame, imposing the *relative* distance of the feet to remain constant. This results in their *relative velocity* to be zero.

Remark: Since we want to be able to change the robot *height*, we want that, of these velocity vectors, only the component along the lines connecting the feet, is constrained to be zero. Therefore, to set the *stance* constraints, we compute the relative velocity vector between two stance feet (e.g. $\dot{\tilde{x}}_i = \dot{x}_{f_i}^r - \dot{x}_{f_{i-1}}^r$) and project it along the direction of the vector connecting two supporting feet (e.g. $\tilde{x}_i = x_{f_i}^r - x_{f_{i-1}}^r$):

$$\underbrace{\begin{bmatrix} (x_{f_1}^r - x_{f_2}^r)^T (\dot{x}_{f_1}^r - \dot{x}_{f_2}^r) \\ (x_{f_2}^r - x_{f_3}^r)^T (\dot{x}_{f_2}^r - \dot{x}_{f_3}^r) \\ (x_{f_3}^r - x_{f_1}^r)^T (\dot{x}_{f_3}^r - \dot{x}_{f_1}^r) \end{bmatrix}}_{\tilde{X}\dot{\tilde{X}}} = \underbrace{\begin{bmatrix} \tilde{x}_1^T (J_1 - J_2) \\ \tilde{x}_2^T (J_2 - J_3) \\ \tilde{x}_3^T (J_3 - J_1) \end{bmatrix}}_{\tilde{J}} \dot{q}^r = 0 \tag{4}$$

where $\dot{q}^r$ are the joint velocities, $J_i \in \mathbb{R}^{3\times 12}$ are the feet Jacobians and $\tilde{J} \in \mathbb{R}^{3\times 12}$ is the Jacobian matrix mapping joint velocities into feet relative velocities $\tilde{X}\dot{\tilde{X}}$. Because feet *virtual* forces are the unknowns in our least-square (LS) problem, we rephrase these constraints in their terms, by exploiting (1):

$$\tilde{J} D_r^{-1} J_{st}^T f_{st}^r = 0 \tag{5}$$

where $J_{st} \in \mathbb{R}^{9\times 12}$ is the stance Jacobian that maps feet forces into joint torques and $D_r \in \mathbb{R}^{12\times 12}$ is the joint *virtual* damping matrix that maps joint velocities into joint torques. Note that $\tilde{J}$ is computed for the reflex-corrected joint configuration q^r computed at the previous control loop.

Then, rewriting the 6 equations (3) in linear form, and stacking them with the 3 stance constraints (4), we can get a unique solution solving for the stance leg *virtual* forces as $f_{st}^r = A^{-1} b$:

$$\underbrace{\begin{bmatrix} I_{3\times 3} & I_{3\times 3} & I_{3\times 3} \\ [x_{f_1}^r - x_{f_{sw}}^r]_\times & [x_{f_2}^r - x_{f_{sw}}^r]_\times & [x_{f_3}^r - x_{f_{sw}}^r]_\times \\ \multicolumn{3}{c}{\tilde{J} D_r^{-1} J_{st}^T} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} f_1^r \\ f_2^r \\ f_3^r \end{bmatrix}}_{f_{st}^r} = \underbrace{\begin{bmatrix} -f_{sw}^r \\ m^r \\ 0_{3\times 1} \end{bmatrix}}_{b} \tag{6}$$

where $[.]_\times$ is the skew symmetric operator associated to the cross-product. The orientation of the robot can be regulated by means of a virtual torsional spring attached at the base origin that adds a non-zero moment m in the momentum equation (3). The variable m^r represents an additional degree of freedom to allow a change in the trunk orientation during the squatting motion since it results in different set of force in the LS solution. The restoring moment is $m^r = -K_{base} e_b$ where $K_{base} \in \mathbb{R}^{3\times 3}$ is a diagonal matrix setting the torsional

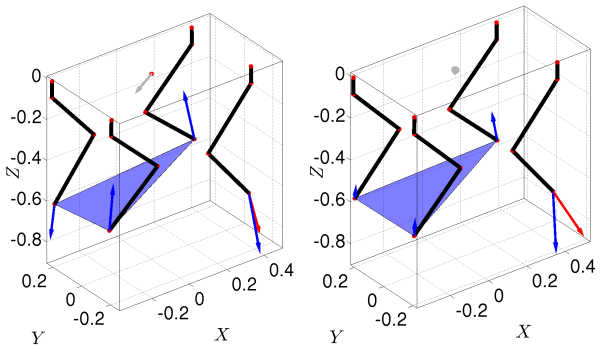

Fig. 6: Different LS solutions (f^r, blue vectors) for different values of the base torsional spring $K_{base_{ii}}$: (a) 100, (b) 1500 $[Nm/rad]$ when the *RF* leg is swinging. f^r_{sw} vector is depicted in red. The base orientation error e_b is in grey while the terrain plane is shaded in light blue. *Remark*: without any loss of generality, just in the Matlab implementation, the base is considered fixed and the terrain plane moves while in the real implementation it is the opposite.

stiffness of the base spring, while e_b is the orientation error with respect to the value of base orientation sampled at the beginning of the swing phase. In particular, if $m^r = \vec{0}$ the LS solution might result in the torso tipping too much in the direction of the swing leg. We can regulate the tipping behaviour by setting the stiffness of the spring that will try to keep the trunk orientation constant (Fig. 6 shows different solutions for different values of the base spring K_{base} when the *RF* leg is swinging). The base orientation error e_b, reduces if the stiffness K_{base} increases. In the case of $K_{base} = diag(100, 100, 100)$ $[Nm/rad]$ the *LH* leg will extend instead of retracting (e.g. to aid the *RF* swing motion). This results in a significant change in the base orientation that might not be desirable. During the reflex action (e.g. squatting), a change in the base orientation can be seen as an opposite change of the inclination of the terrain plane (plane fitting trough the 3 stance feet, see Fig. 5). Therefore, it is convenient to consider the normal of the terrain plane n_t for the computation of the orientation error. This can be obtained by considering the relative distance vectors $\tilde{x}_1$ and $\tilde{x}_2$ between two (arbitrary) stance feet, as:

$$n_t = \frac{\tilde{x}_1 \times \tilde{x}_2}{|\tilde{x}_1 \times \tilde{x}_2|} \tag{7}$$

The error vector between n_t and its desired value (sampled at the beginning of the swing phase) n_t^d can be computed by simple geometric relations:

$$\begin{cases} \hat{a}_t = n_t \times n_t^d \\ \delta = acos(n_t^\top n_t^d) \\ e_t = \delta \hat{a}_t \end{cases} \tag{8}$$

where $\hat{a}_t$ is the axis of the finite rotation to bring n_t onto n_t^d. Then for the computation of the base moment we set $e_b = -e_t$.

C. End Stop Repulsion

The height reflex can also take care of stopping the joint motion, in a smooth way, whenever a kinematic limit is reached. The end-stop limits are modelled as virtual springs at the joint level that generate virtual torques $\tau_{kl} \in \mathbb{R}^{12}$ that oppose the joint motion when it is close to its range limits. The reflex will still use the other joints to redistribute the swinging motion. The torsional springs become active when the joint positions are close (within a certain user-defined angular distance Δ) to the limits.

$$\begin{aligned} \tau_{kl} = &\mathbf{1}(q > q_{max_{th}}) K_{kl}(q_{max_{th}} - q_i) + \\ &\mathbf{1}(q < q_{min_{th}}) K_{kl}(q_{min_{th}} - q_i) \end{aligned} \tag{9}$$

where $q_{max}, q_{min} \in \mathbb{R}^{12}$ are the joint range upper and lower bounds, respectively. $q_{max_{th}} = q_{max} - \Delta$ and $q_{min_{th}} = q_{min} + \Delta$ represent the joint thresholds for which the virtual springs become active, and $\mathbf{1}(.)$ is the associated indicator function. $K_{kl} \in \mathbb{R}^{12\times12}$ is the end stop virtual springs stiffness matrix. This resembles a "virtual strip" that pulls the joint away when it is approaching the kinematic limits, thus stopping the motion (if the stiffness is high enough).

D. Viscosity Based Planning

The solution f^r_{st} obtained in (6) together with τ_{kl} computed in (9) are given as input to the *virtual* damper model of the robot (1). The output will be the motion of all the *virtual* joints (of the simplified model), according to the viscosity D_r that was set, and the kinematic limits. In particular, the resulting motion will be dependent on the action of two torques: one is due to the Cartesian forces f^r(mapped at the joint space) that tries to achieve the swing motion (either moving the swing leg and/or lowering the body) and another due to the spring $K_{kl_{jj}}$ which prevents to reach the kinematic limit. When these two torques are in equilibrium then the joint stops.

$$\dot{q}^r = D_r^{-1} \underbrace{(J(q^r)^T f^r - \tau_{kl})}_{\tau^r} \tag{10}$$

where $J \in \mathbb{R}^{12\times12}$ is the Jacobian matrix for all the feet. Since the output of the damping model is a vector of velocities, at each control loop, we integrate it (with a trapezoidal rule) to obtain the reflex-corrected position references q^r. Then, through forward kinematics, we compute the new vector of feet positions/velocities x^r_f, $\dot{x}^r_f$ *corrected* by the *height reflex*, that will be sent (in place of the original plan) to the controller (see Fig. 2):

$$\begin{cases} x^r_f = \mathbb{K}(q^r) \\ \dot{x}^r_f = J(q^r)\dot{q}^r \end{cases} \tag{11}$$

where $\mathbb{K}(.)$ is the non linear function computing the feet kinematics from joint angular positions.

E. Damping Scheduler

A smart way to spread the swing motion among the legs is to set different virtual damping values for the joints of each leg. The idea is that, by setting a lower viscosity (e.g.

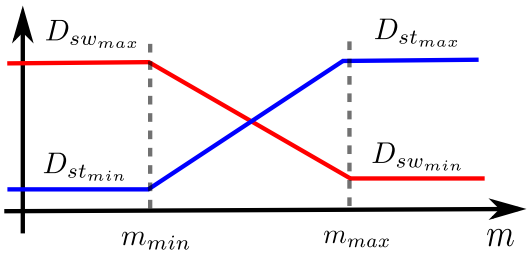

Fig. 7: Damping schedule for swing and stance legs along with the swing leg mobility represented by the velocity transformation ratio m.

damping) for all the joints of a leg, that leg will move more. Our heuristic is to regulate the damping according to the swing leg mobility (see Fig. 7). When the swing leg is losing mobility we gradually increase the damping for the joints of the swing leg, while, in parallel, we reduce it for the joints of the remaining stance legs, in order to *redistribute* the motion between them.

We decided to use the *velocity transformation ratio m* [17] as a metric to estimate the mobility of the swing leg (for the computation of m refer to Section III-F). When the swing leg mobility is high ($m > m_{max}$) we set the damping very high for the stance legs and very low for the swing leg such that the reaching motion will be entirely responsibility of the swing leg joints, as in usual operation. When the mobility starts to decrease below m_{max}, the damping is increased (up to $D_{sw_{max}}$) on the swing leg and decreased (up to $D_{st_{min}}$) on the stance legs.

Remark. Since in the upward phase of the swing the leg is usually increasing mobility, in this work we restrict the activation to the downward phase of the swing.

F. Velocity Transformation Ratio

It is well known that close to singularity a manipulator can move very slowly and mobility is lost. Having mobility is important to keep the joint velocities in a reasonable range and to ensure reachability of the target. Different metrics can be used to obtain a scalar representative of the leg mobility [18], which represent a distance from the singular configuration (in the HyQ case when the leg is completely stretched).

$$\begin{cases} w & = \sqrt{det(J_f J_f^T)} \\ \kappa^{-1} & = \frac{\sigma_{min}(J_f(q^r))}{\sigma_{max}(J_f(q^r))} \\ m(v) & = (v^T (J_f J_f^T)^{-1} v)^{-0.5} \end{cases} \quad (12)$$

where $J_f \in \mathbb{R}^{3\times 3}$ is the foot Jacobian computed at the joint configuration q^r. w is the *manipulability measure* that represents the volume of the manipulability ellipsoid. It gives an idea of the global manipulation capability of the leg. However, this number depends on the kinematics of the leg (e.g. leg length) and has limited accuracy when close to the singularity, because it can change several orders of magnitude. Another possibility is to use the κ^{-1}, that is the

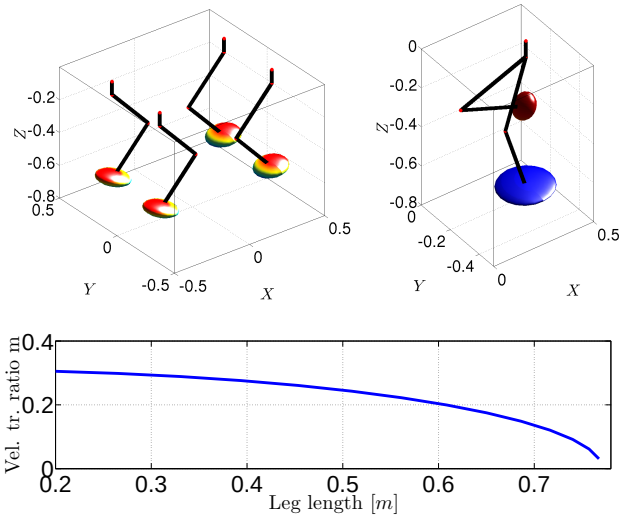

Fig. 8: (top left) Velocity ellipsoids computed for a default configuration, (top right) RF leg. Velocity ellipsoids for a retracted and extended configuration. (lower) Plot of the velocity transformation ratio with the leg extension.

reciprocal of the condition number (ratio between maximum and minimum singular value) of J_f which is independent from the volume of the ellipsoid and it gives a measure of the eccentricity [18]. However, this is more computationally expensive, because it involves matrix decomposition to compute the singular values. Since we are interested in the mobility loss in a particular *direction*, we choose to evaluate the *velocity transformation ratio* [17], [19] $m(v)$ along a specific direction v. This is equal to the distance from the center to the surface of the velocity ellipsoid $\dot{x}(J_f J_f^T)^{-1}\dot{x} = 1$ along the directional vector v. In our case we consider as v the terrain normal n_t, that is the direction along which the haptic searching motion occurs and searches for the contact [6]. Figure 8 (top left) shows normalized velocity ellipsoids computed for a default robot configuration, while (top right) shows the change of the velocity ellipsoid for the *RF* leg when passing from a retracted to an extended configuration. The lower plot shows how the leg mobility changes with the leg extension.

IV. Results

We present in this section experimental results [7] both in simulation and on the real hardware of our robot HyQ stepping down from platforms. In our experiments we found a satisfactory behaviour for the height reflex parameters that are summarized in Table I. All the experiments are conducted with HyQ, our 80 kg hydraulically actuated quadruped robot [13], [21]. The HyQ system has 12 active Dofs and it is fully-torque controlled. It is equipped with joint encoders for joint position measurement and an Inertial Measurement

[6] Attention should be paid when mixing orientation mobility with linear mobility in the computation of m [20]. However, this is not an issue for HyQ because, since it has point feet, we are considering only the linear case.

[7] A video of the experiments can be found at youtu.be/FPkvu29WLUc.

TABLE I: Height Reflex Parameters

Parameter	Symbol	Value
Cart Stiffness $[Nm/rad]$	K_x	diag(10^3[5,5,5])
Base Spring Stiffness $[Nm/rad]$	K_{base}	1500
End Stop Stiffness $[Nm/rad]$	K_{kl}	1000
End Stop influence $[rad]$	Δ	0.05
Min/max stance damp.$[Nms/rad]$	$[D_{st_{min}} D_{st_{max}}]$	50, 1000
Min/max swing damp. $[Nms/rad]$	$[D_{sw_{min}} \; D_{sw_{max}}]$	5 100
Min/max transm. ratio [1]	m	0.19 0.21

Unit (MicroStrain) to measure the trunk orientation. We performed several experiments with the robot stepping down from a (a) 16 *cm*, (b) 21 *cm*, (c) 24 *cm* high pallet (see Fig. 1). We repeated the experiments with the height reflex enabled and disabled (exploiting only terrain adaptation capabilities). The forward velocity was set to 0.1 m/s. Having the reflex enabled improved the overall robot mobility by 10% for experiment (a)[8]. However, for experiments (b) and (c), the height reflex was fundamental for the success of the task, because the swing leg was getting very close to the singularity. According to our crawl strategy we trigger the stance when approaching the end-stop limits, to prevent the state machine from getting "stuck" on the swing phase. Figure 9 shows experimental data when the reflex is in action on the *RF* leg (e.g. when this leg is swinging down from the step). The upper plot shows the *height reflex* activation during the swing down, while the second plot shows the scheduling of the damping according to the mobility loss of the swing leg. The 3^{rd} and 4^{th} plot show how the feet trajectories are modified by the reflex for the swing leg and the stance legs respectively: part of the swing motion is redirected to the stance legs that retract leading to the trunk "squatting" motion. Initially most of the motion is in the knee but when the leg gets stretched the viscosity is magnified at the end effector and the leg naturally moves less and the motion is transferred to the torso.

Figure 10 shows a time interval of the experiment (b) where the kinematic limits are reached for the hip joint of the *RH* leg. The upper plot shows that the joint trajectory does not cross the limit. Furthermore, thanks to the presence of a virtual spring the joint will never approach the limit with non-zero velocity (middle plot). The lower plot shows how the foot trajectory is corrected to have the kinematic limit satisfied. Finally Fig. 11 shows the variation of the robot pitch during the whole (b) experiment. In the first half the pitch is increasing while the robot is stepping down, while in the second one the trunk is going back to be horizontal (e.g. zero pitch).

Remark 1. A simple clipping of the reference joint position trajectory does not ensure that the velocity would be zero at the limit. Depending on the joint inertia and the controller

[8]The overall mobility was computed as $\bar{m} = \sum_{k=1}^{N} \sum_{i=1}^{4} c_{sd}(k) m_i(k)$ where $c_{sd}(k) \in \{0,1\}$ represents the swing down phase, while N is the total length of the dataset.

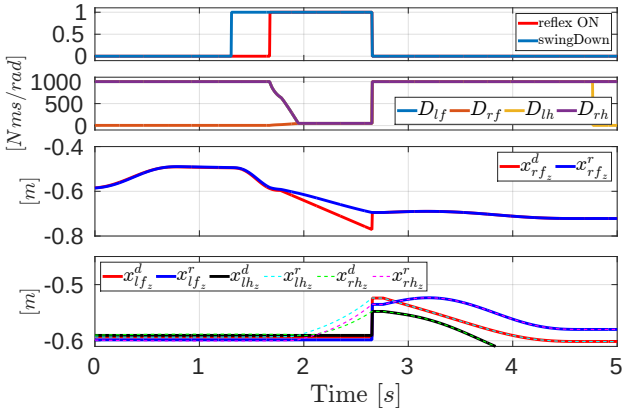

Fig. 9: Experimental data from (b) experiment (21 *cm* platform) when the reflex is in action on the *RF* (swinging) leg. The upper plot is reflex activation, the second plot shows the scheduling of the damping according to the mobility loss of the swing leg. The last two plots show the feet trajectories modified by the reflex for the swing leg (third) and the stance legs (fourth).

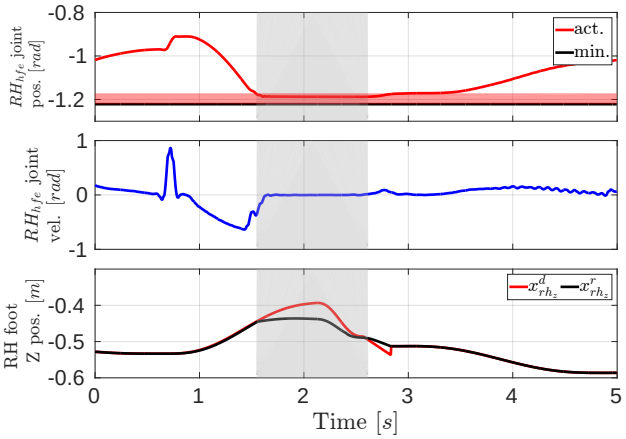

Fig. 10: Experimental data from (b) experiment (21 *cm* platform). The upper plot shows that a kinematic limit is approached on the RH_{hfe}. The grey shaded area shows the time interval in which the joint enters in the joint range (red shaded area) where the end-stop virtual spring is active. The middle plot shows that the joint velocity is driven to zero before approaching the end-stop. The lower plot shows how the *RH* foot trajectory is corrected.

bandwidth, this residual velocity can create impacts with the end-stops. These are undesirable, because they can result in errors in the estimation of the ground reaction forces or in the controller.

Remark 2. From the actuation point of view, the torques requested by the replanned trajectory are not very different from the ones of the default plan. In any case, if an appropriate weight matrix is set in the cost function [12], the trunk controller can always find a solution that minimizes the torque vector norm.

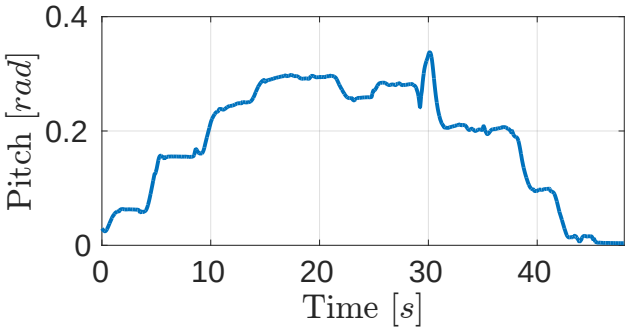

Fig. 11: Trunk pitch variation during the (b) experiment.

V. Conclusions

We proposed a computationally efficient *online* replanning strategy for legged robots that locomote on rough terrain. The strategy is called *height reflex* and it is important to address big terrain changes like stepping down from a high platform. This is a module that will become part of the reactive features in the locomotion framework of our HyQ robot. The *height reflex* solves the issue of losing mobility when the swing leg searches the contact by spreading the motion, in a kinematially consistent way, onto the other supporting legs, while preserving static stability. We demonstrated both in simulation and real experiments that the height reflex makes the difference for big steps (e.g. above 16 *cm*). In the case of moderate step height, the height reflex still enhances the locomotion because it prevents the legs from stretching too much. In future works we are planning to test the *height reflex* on the HyQ2Max robot [22] which has bigger joint range mobility, to address even higher steps.

Acknowledgement

This work was supported by Istituto Italiano di Tecnologia (IIT), with additional funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 601116 as part of the ECHORD++ (The European Coordination Hub for Open Robotics Development) project under the experiment called *HyQ-REAL*.

References

[1] C. Mastalli, M. Focchi, H. Ioannis, A. Radulescu, S. Calinon, J. Buchli, D. G. Caldwell, and C. Semini, "Trajectory and Foothold Optimization using Low-Dimensional Models for Rough Terrain Locomotion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[2] A. Winkler, C. Mastalli, M. Focchi, D. G. Caldwell, and I. Havoutis, "Planning and Execution of Dynamic Whole-Body Locomotion for a Hydraulic Quadruped on Challenging Terrain," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5148–5154, 2015.

[3] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, "Learning, planning, and control for quadruped locomotion over challenging terrain," *The International Journal of Robotics Research*, vol. 30, pp. 236–258, nov 2010.

[4] M. Fallon, P. Marion, R. Deits, T. Whelan, M. Antone, J. McDonald, and R. Tedrake, "Continuous humanoid locomotion over uneven terrain using stereo fusion," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pp. 881–888, nov 2015.

[5] M. Camurri, M. Fallon, S. Bazeille, A. Radulescu, V. Barasuol, D. G. Caldwell, and C. Semini, "Probabilistic Contact Estimation and Impact Detection for State Estimation of Quadruped Robots," *IEEE Robotics and Automation Letters*, vol. 2, pp. 1023–1030, apr 2017.

[6] M. Bloesch, M. Hutter, M. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, "State Estimation for Legged Robots-Consistent Fusion of Leg Kinematics and IMU," *Robotics: Science and Systems*, 2012.

[7] X. Xinjilefu, S. Feng, W. Huang, and C. G. Atkeson, "Decoupled state estimation for humanoids using full-body dynamics," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 195–201, 2014.

[8] V. Barasuol, M. Camurri, S. Bazeille, D. Caldwell, and C. Semini, "Reactive Trotting with Foot Placement Corrections through Visual Pattern Classification," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[9] M. Focchi, V. Barasuol, I. Havoutis, C. Semini, D. G. Caldwell, V. Barasul, and J. Buchli, "Local Reflex Generation for Obstacle Negotiation in Quadrupedal Locomotion," *International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR)*, pp. 1–8, 2013.

[10] G. Orlovsky, D. T.G., and S. Grillner, *Neuronal Control of Locomotion: From Mollusc to Man*, vol. 3. Oxford University Press, 1999.

[11] J. Albiez, W. Ilg, T. Luksch, K. Berns, and R. Dillmann, "Learning a reactive posture control on the four-legged walking machine BISAM," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 2, pp. 999–1004 vol.2, 2001.

[12] M. Focchi, A. Del Prete, I. Havoutis, R. Featherstone, D. G. Caldwell, and C. Semini, "High-slope terrain locomotion for torque-controlled quadruped robots," *Autonomous Robots*, pp. 1–14, 2016.

[13] 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*, 2011.

[14] D. Bellicoso, C. Gehring, J. Hwangbo, P. Fankhauser, and M. Hutter, "Perception-Less Terrain Adaptation through Whole Body Control and Hierarchical Optimization," in *IEEE-RAS International Conference on Humanoid Robots*, 2016.

[15] P. Marion, R. Deits, L. Manuelli, M. Antone, H. Dai, T. Koolen, J. Carter, M. Fallon, S. Kuindersma, and R. Tedrake, "Director : A User Interface Designed for Robot Operation With Shared Autonomy," 2016.

[16] C. Liu, C. G. Atkeson, S. Feng, and X. Xinjilefu, "Full-body motion planning and control for the car egress task of the DARPA robotics challenge," *IEEE-RAS International Conference on Humanoid Robots*, vol. 2015-Decem, pp. 527–532, 2015.

[17] S. Chiu, "Control of redundant manipulators for task compatibility," *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, vol. 4, pp. 1718–1724, 1987.

[18] L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulators*. Springer, 2000.

[19] T. Yoshikawa, "Analysis and Control of Robot Manipulators with Redundancy," 1984.

[20] D. Prattichizzo and A. Bicchi, "Dynamic analysis of mobility and graspability of general manipulation systems," *IEEE Transactions on Robotics and Automation*, vol. 14, pp. 241–258, apr 1998.

[21] C. Semini, V. Barasuol, T. Boaventura, M. Frigerio, M. Focchi, D. G. Caldwell, and J. Buchli, "Towards versatile legged robots through active impedance control," *The International Journal of Robotics Research*, pp. 1–18, 2015.

[22] C. Semini, V. Barasuol, J. Goldsmith, M. Frigerio, M. Focchi, Y. Gao, and D. Caldwell, "Design of the Hydraulically-Actuated, Torque-Controlled Quadruped Robot HyQ2Max," *IEEE/ASME Transactions on Mechatronics*, vol. PP, no. 99, p. 1, 2016.