

Dynamic Trotting on Slopes for Quadrupedal Robots

Christian Gehring*, C. Dario Bellicoso*, Stelian Coros[†],

Michael Bloesch*, Péter Fankhauser*, Marco Hutter*, and Roland Siegwart*

*Autonomous Systems Lab, ETH Zurich, Switzerland (gehrinch@ethz.ch) [†]Carnegie Mellon University, USA

Abstract—Quadrupedal locomotion on sloped terrains poses different challenges than walking in a mostly flat environment. The robot's configuration needs to be explicitly controlled in order to avoid slipping and kinematic limits. To this end, information about the terrain's inclination is required for carefully planning footholds, the pose of the main body, and modulation of the ground reaction forces. This is even more important for dynamic trotting, as only two support legs are available to compensate for gravity and drive a desired motion. We propose a reliable method for estimating the parameters of the terrain quadrupedal robots move on, in the face of limited perception capabilities and drifting robot pose estimates. By fusing inertial measurements, kinematic data from joint encoders and contact information from force sensors, the local inclination can be robustly estimated and used to optimize the contact forces to reduce slippage. The estimated terrain information, namely the pitch and roll angles of the ground plane, is exploited in an extended version of our previous model-based control approach. Our improved control framework enabled StarLETH, a medium-sized, fully autonomous, torque-controllable quadrupedal robot, to trot on slopes of up to 21° .

I. INTRODUCTION

Mobile robots require a great deal of maneuverability, agility and skill in order to operate in every-day environments. Due to their potential ability of overcoming gaps, steps, slopes, and other types of obstacles, legged robotic platforms are particularly promising. However, the motor capabilities of state-of-the-art legged robots still fall far behind those of living animals, who can negotiate even the most precipitous environments with grace [1]. Among quadrupedal robots, the ones that come closest to this type of performance are the Boston Dynamics BigDog [2] and LS3, which were shown trotting uphill and on rocky slopes. Unfortunately, no experimental data is available for these feats, and very little is known about the control strategies that are employed.

As reported in biomechanics [1], locomotion on slopes requires different strategies than on flat ground. The posture and swing leg trajectories are adapted to the terrain as well as the ground reaction forces. Inspired by these insights, we extended our locomotion control framework in order to enable StarLETH, a quadrupedal robot, to dynamically trot up and down significant slopes of up to 20° . In this paper, we discuss both a method for accurately detecting the slope of the terrain the robot is walking on, as well as a necessary set of adaptations to the control strategies employed by our robot.

This research was supported by the Swiss National Science Foundation through the National Centre of Competence in Research Robotics.

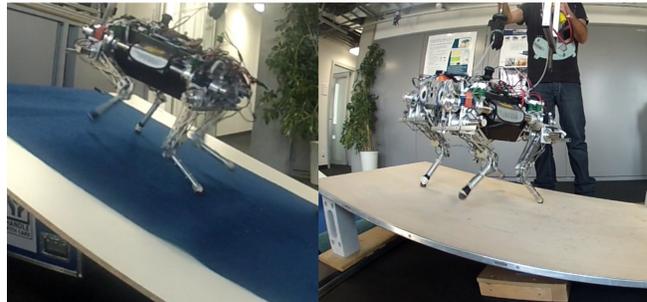


Fig. 1. Our quadrupedal robot, StarLETH, employing a dynamic trotting gait to move forward and turn on slopes of up to 21° (38%) with only two legs simultaneously in contact with the ground.

A. Related Work

Walking on slopes has been a topic of active interest since the early days of legged robotics [3], [4], and it continues to be heavily researched in recent years [5], [6], [7], [8], [9]. Most of the existing body of work on this topic deals with statically stable gaits. This largely simplifies the task since the robot has always at least three feet in contact with the ground such that the slope can be determined at each time step from the latest measurements during the whole gait cycle. In contrast, we investigate the challenge of controlling dynamic trotting on sloped terrains.

Generally speaking, control strategies for locomotion on sloped terrains can be divided into two categories: methods which include terrain properties during *motion planning* and *reflex-based* methods which treat variable slopes in the environment as perturbations. The latter class of methods bypass the problem of explicitly modeling the environment, but have only been shown to work on relatively modest slopes [10], [11]. Methods in the first category either have perfect knowledge about the environment – information that can be used for planning footholds, as demonstrated on LittleDog [8], or estimate a rough set of parameters that can be used to model the terrain. This approach was recently employed to enable a hexapedal robot, Lauron, to perform static walking on slopes [9]. In this work, the slope of the terrain was estimated by fitting a plane through the support legs. Since at least three legs were in contact with the ground at any moment in time, this approach worked reliably. However, for gaits of increased agility, such as the quadrupedal trot where only two legs are simultaneously in contact with the ground for extended periods of time, a more sophisticated approach is needed, as described in this paper.

After obtaining a model of the environment, a very impor-

tant step is to adapt the signals output by the controllers in an appropriate manner. One common strategy is to adapt the roll and pitch angle of the robot with respect to the slope. The configuration of the legs can also be adapted. For instance, the extension of the legs can be adapted using neural control systems [12], [5], and for robot legs exhibiting kinematic redundancies, the orientation of the leg segments with respect to the vertical direction can also be controlled, as executed by Lauron with its additional joint [9].

In addition to controlling kinematic parameters such as the pose of the body or the configuration of the legs, *torque-controlled* robots have the important ability of precisely modulating the ground reaction forces applied by the stance feet, such that slipping is reduced or eliminated altogether [13], [8]. In fact, internal force directions among the legs allow changing the ground reaction forces without influencing the robot motion. Our adapted control strategy combines many of these aspects within one unified control framework: the roll and pitch angles of the robot's body are modified based on the estimated terrain slope, the location of the swing feet are controlled so as to minimize early or late contacts with the ground while ensuring a telescoping posture [1], and the contact forces generated by the stance legs are appropriately modulated to control the overall motion of the robot.

B. Contribution

The quadruped locomotion control framework we introduced earlier [10] was successfully tested on the torque-controlled quadruped StarIETH [14]. The control method can cope with limited sensory information and leads to a large number of gaits, such as forward and lateral walks, trot, prnk and bound as shown in [15]. The controllers are robust against unanticipated perturbations caused by pushes or irregular terrain, but they fail on unperceived slopes larger than 5° due to the inherent (and wrong) assumption that the ground upon which the robot moves is flat. The goal of this work is to extend our previous control framework in order to enable dynamic quadruped locomotion on significantly steeper slopes.

The contribution of our work is twofold. First, we propose an improved model for computing the slope of the terrain the robot is moving on, and we explain how to deal with the drift that inevitably occurs in estimating the state of the robot. Second, we describe the set of controller adaptations that were needed in order to allow the robot to trot on steep slopes. We validate our approach with a variety of experiments, both in simulation and on the quadrupedal robot StarIETH.

II. ESTIMATION OF THE TERRAIN

In order to correctly adapt the motion of the robot as it moves across sloped terrains, it is imperative that a model of the environment is first built. This task is complicated by the limited perception capabilities of our robot. In particular, the robot is equipped with optical force sensors at each foot, allowing contacts with the environment to be accurately detected. The only other available sensors consist of joint

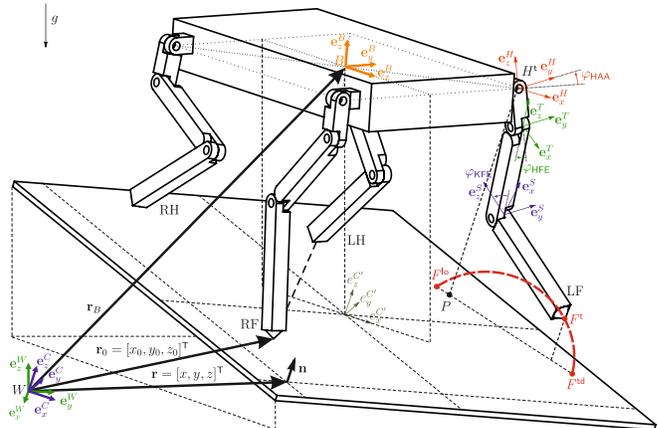


Fig. 2. Model of the robot and the sloped terrain: world frame W , base frame B , control frame C , shifted control frame C' , joint positions φ_i , base position \mathbf{r}_B , plane parameters (\mathbf{r}, \mathbf{n}) , right fore (RF) stance foot on plane \mathbf{r}_0 , hip location H^t , projected hip location on slope \mathbf{P} (along \mathbf{n}), foot location F^t , foot location at lift-off F^{lo} , and foot location at touch-down F^{td} .

encoders and an inertial measurement unit (IMU). The data from these three types of sensors are fused through an extended Kalman filter [16] to estimate the pose and linear and angular velocities of the torso with respect to a world frame W (cf. fig. 2), which are required to properly control the whole-body. The filter fuses the IMU measurements with estimates based on the leg configurations, which are measured relative to the main body by joint encoders. By taking into account the feet positions of the grounded legs, which are assumed to have zero velocity, the linear velocity of the body becomes observable. We note that this approach does not need any assumption about the terrain and that it should provide more accurate estimates than conventional complementary filters implemented on commercially available IMUs.

The lack of vision or extrinsic sensors means that the position \mathbf{r}_B and yaw angle (heading) of the robot with respect to the world frame are not directly observable, and thus *drift* over time. This affects the foot placement on the terrain, which needs to be planned in a consistent (world) coordinate system. As a consequence, this drifting effect needs to be compensated to some extent by a proper selection of a model of the ground and an adequate parameter estimation.

We model the terrain using a plane whose parameters are estimated using the limited sensory information that is available to the robot. The plane parameters can be trivially computed as long as at least three legs are in contact with the ground as shown in [9]. However, for the trotting gait that we consider, *only two* legs are commonly in contact with the ground at any moment in time. The plane is consequently not well defined. We therefore propose to employ a *history* of foot holds in order to estimate the slope of the terrain.

A. Model and Parameter Estimation

A point $\mathbf{r} = [x, y, z]^T$ on a plane can be described by the following equivalent equations:

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

where $\mathbf{r}_0 = [x_0, y_0, z_0]^T$ is a point on the plane and $\mathbf{n} = [a, b, c]^T$ is the normal of the plane (cf. fig. 2).

There are in total four parameters (a, b, c and d) which need to be estimated from the foot hold measurements. By assuming that $c = 1$ i.e. the plane can never be equal to any plane with the normal perpendicular to the z -axis of the world frame, the height of the foot position measurements $\mathbf{r}_i = [x_i, y_i, z_i]^T$ on the plane becomes a linear function of the parameters $\boldsymbol{\pi} = [a, b, d]^T$:

$$z_i = [-x_i \quad -y_i \quad 1] \begin{bmatrix} a \\ b \\ d \end{bmatrix} \Rightarrow \zeta_i = \mathbf{h}_i \boldsymbol{\pi}. \quad (2)$$

A simple way to estimate $\boldsymbol{\pi}$ is to solve a least-squares problem: $\hat{\boldsymbol{\pi}} = \mathbf{H}^+ \boldsymbol{\zeta}$, where \mathbf{H}^+ is the pseudo-inverse of \mathbf{H} . The dimension of $\boldsymbol{\pi}$ implies that at least three measurements are required. For dynamic gaits, considering only the current support legs, as in static gaits, is therefore not enough. Since the estimated position of the robot \mathbf{r}_B drifts over time, the feet positions $\mathbf{r}_i = f(\mathbf{r}_B, \mathbf{R}_B, \boldsymbol{\varphi})$ expressed in world frame, which are a function of the estimated robot's pose ($\mathbf{r}_B, \mathbf{R}_B$) and joint angles $\boldsymbol{\varphi}$, drift as well. We therefore consider only the most recent position estimates for the grounded feet, instead of their locations at touch-down. For the swing legs, we take the last position before lift-off into account.

We note that when a foot touches the ground, a contact is detected before the compliant foot is fully compressed. To filter out this effect and measurement noise, the normal \mathbf{n} and the point \mathbf{r} on the plane, which are updated with 400 Hz, are low-pass filtered with a cut-off frequency of 3 Hz.

Note that these implementation details are important to make the method work on a real system with sensor noise, modelling errors, and hidden states.

B. Evaluation

To validate the proposed method, we ran several experiments where the physical robot is tasked with trotting on different slopes. Figure 3 shows the results of an experiment where the robot traverses a flat terrain first, and then walks uphill on a slope with angle $\theta = 21^\circ$ and finally walks in place on the slope. To validate the estimated parameters, we want to compare the estimated slope angle with the ground truth which we measured one. For this reason we visualize the pitch $\hat{\theta} = \arctan \frac{a}{c}$ and the roll angle $\hat{\phi} = \arctan \frac{b}{c}$ in fig. 3c). As long as the heading direction of the robot is aligned with the plane, the pitch angle should correspond to the slope angle. The mean of the estimated pitch angle over the time window [20, 30s], when the robot is walking on the slope, is $21.2^\circ \pm 0.2^\circ$. For an experiment on a 10.2° -slope, the estimates indicated a mean angle of $10.4^\circ \pm 0.1^\circ$.

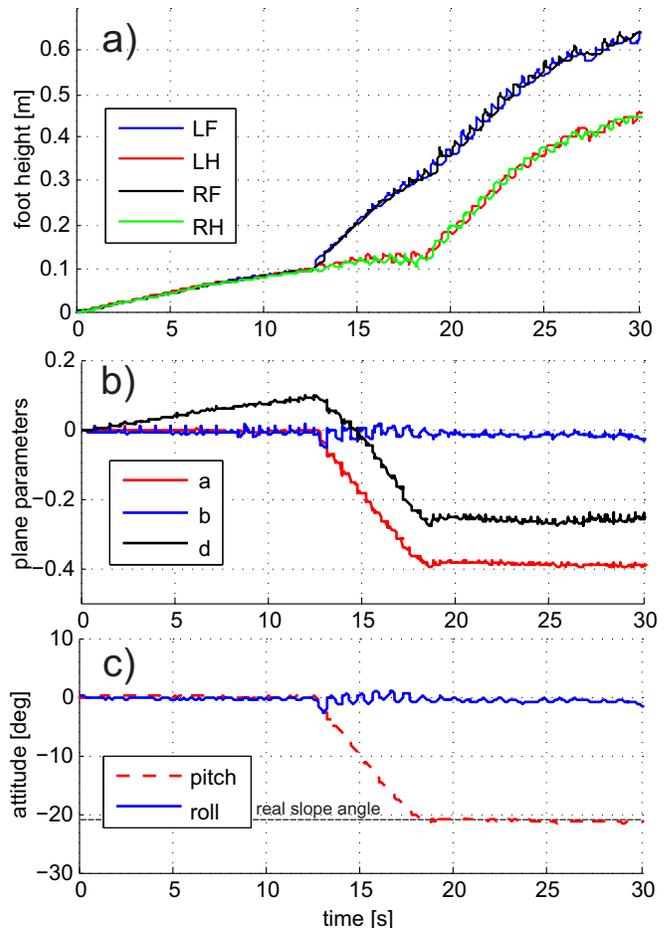


Fig. 3. a) shows the measured foot heights in world frame (LF: left fore, RF: right fore, LH: left hind, RH: right hind), which are used for estimating the plane parameters as shown in b). The attitude angles in c) visualize the terrain slope.

The drift of the foot heights can be clearly seen in fig. 3a). Nevertheless, our method can compute the terrain slope very accurately. An added benefit of our approach is that, as the robot begins to climb the platform, we get a smoothly varying estimate that interpolates between the flat region and the actual parameters of the terrain.

III. LOCOMOTION ON FLAT TERRAIN

The locomotion controller we presented in prior work [10] is based on a few key assumption about the robot and its environment. In particular, without the method we described for estimating the slope of the terrain, one major assumption is that the environment is *flat*. While this simplifies somewhat the control problem, it also makes locomoting on even modest slopes all but impossible. In this section, we briefly summarize our previous control approach and we identify the main reasons for its limited ability to correctly handle sloped terrain.

To begin with, we employ a *gait pattern* as shown in fig. 4a) to output a prescribed sequence of footfalls and provide coordination between the limbs. At any moment in time, each leg is therefore either used in stance mode, where its role is to support the main body, or in swing mode, where

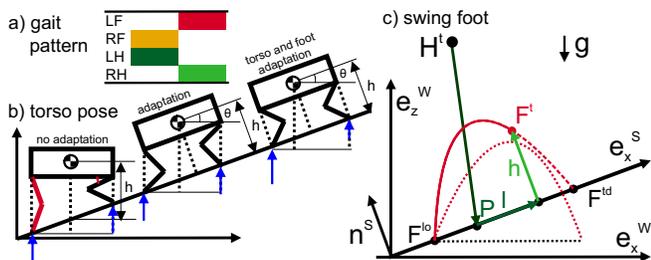


Fig. 4. a) shows the gait pattern. The bars indicate when each leg should be grounded. b) illustrates different strategies for the pose of the robot on slopes with the contact forces in blue. c) shows the swing foot trajectory composition that is aligned with the slope as solid red curve and the wrongly aligned one as dotted curve.

its role is to move towards the next stepping location. The desired trajectory of the robot’s body is defined over the full cycle of a given gait. Without an explicit model of the terrain, this trajectory is pre-defined. As outlined in the next section, this design decision already provides a limiting factor for the steepness of the slope the controller can negotiate.

At any moment in time, the desired pose of the torso can only be controlled indirectly, through the support legs which are in contact with the ground. As our robot is torque controllable, it can explicitly modulate the ground reaction forces applied through the stance legs. Special care is taken to ensure that the resulting ground reaction forces do not cause the feet to slip. This is done by enforcing unilateral constraints between the normal and tangential components of the ground reaction forces. However, this computation is once again performed without taking the model of the terrain into account, thus limiting the performance of our controllers.

The swing legs are position-controlled so that the feet track a desired trajectory expressed in Cartesian coordinates. The desired trajectory of the swing legs is computed using a combination of feed-forward and feedback contributions. The desired height of the swing feet is pre-programmed using spline functions. To ensure that the robot plants the swing feet at the right moment in time, it needs to have an accurate model of the environment. This causes serious problems when walking on slopes, as early or late contacts can significantly disturb the steady state of the locomotion cycle.

IV. LOCOMOTION CONTROL ON SLOPED TERRAIN

Walking on slopes introduces new challenges as compared to locomoting over flat terrain. Some of these challenges are related to the design of the robot, while others are a function of the gait that is employed. Perhaps the most intuitive restriction related to the design of the robot is the material used for the feet, as it defines the coefficient of friction and therefore limits the maximal tangential force that can be applied. When on a slope, this tangential component becomes crucial even for simple actions like standing upright. As a simple example, with a friction coefficient of $\mu = 0.6$, the maximum slope angle that still allows non-

slipping ground reaction forces that compensate for gravity is $\theta = \arctan(\mu) = 31^\circ$.

The robot’s kinematic design – the dimensions of the body, the location of the hip joints, the topology of the legs and the joint limits – defines the admissible workspace of the legs and has a significant impact on the ability of a robot to locomote on slopes. Fortunately, adjustments to the control strategy can help.

A. Adaptation of the Control Frame

The task of the locomotion controller is to track the desired heading, lateral and turning velocities of the robot. The simplest approach would be to express these velocities in the base frame B shown in fig. 2. However, the torso can undergo a motion such that this coordinate frame is not appropriate to express such inputs. For instance, the main body pitches heavily during a bounding gait. Hence, for a flat terrain, the desired velocities are naturally expressed in a so-called *control frame* C whose z -axis is aligned with gravity and whose x -axis is perpendicular to gravity and oriented according to the heading of the robot. The origin of C is fixed to the origin of the world frame W .

When walking on a slope, we adapt this frame to the slope as illustrated in fig. 2 to keep the trajectories consistent with the perceived terrain. The z -axis of the control frame (e_z^C) is aligned with the normal of the slope (n), whereas the x -axis of the control frame (e_x^C) is parallel to the projected x -axis of the base frame on the slope ($e_x^{C'}$). The origin of the coordinate system C remains fixed to the origin of the world frame W such that only the orientation of the frame is changing over time. The target velocities of the robot together with the desired height with respect to the ground and the orientation of the main body are described in this new frame.

B. Adaptation of the Torso Orientation

Figure 4b) shows three possible configurations of a robot on a sloped terrain. The left-most configuration maintains a horizontal posture for the body. Consequently, the downhill leg is almost fully stretched, thus reaching a nearly singular configuration, while the uphill leg is crouched and close to a joint limit. In contrast, the configuration in the middle of fig. 4b) aligns the orientation of the body to the slope. As a result, both pairs of legs are in a much more natural pose, and significantly farther away from singular configurations or joint limits. We note that this postural adaptation is not possible if the controller assumes the robot is moving across flat terrain. For all our experiments, the robot automatically orients its roll and pitch according to the estimated slope parameters, and aims to keep a fixed body height (0.42 m) relative to the terrain.

C. Adapting the Configuration of the Stance Legs

A closer look at the middle and right-most configurations shown in fig. 4b) reveals that, given the same body orientation, different approaches for planting the stance legs exist. In particular, these two distinct strategies are known in

biomechanics as the *telescoping strut* and *lever mechanism*, respectively [1]. When employing the telescoping strut strategy, the stance feet are positioned directly below the hips, leading to the downhill knee getting potentially too close to the terrain. With the lever mechanism approach, the vector between the hips and the feet is parallel to the normal of the terrain, resulting in a further improved configuration of the stance legs. However, the exact strategy being employed has further implications in controlling the motion of the robot.

The placement of the stance legs relative to the body influences the distribution of the contact forces, which are represented by the blue arrows in fig. 4b). As illustrated, the lever mechanism strategy leads to a larger force being applied to the downhill leg than to the uphill one. This is due to the differently-sized orthogonal components of the forces' moment arms, in conjunction with the need to avoid a pitching moment. However, the trotting gait we employ poses an additional difficulty, as *only diagonally opposite pairs of legs* are in contact with the ground at most points in time. Consequently, the torque about the roll-axis of the robot becomes very challenging to regulate correctly.

To better understand this problem, we investigate in detail the two configurations corresponding to the telescoping strut and lever mechanism respectively. For both cases, the task is to compute the set of slip-free contact forces that maintain the position and orientation of the robot (i.e. zero net force and torque), as shown in Fig 5. The ground reaction forces \mathbf{f}_1^c and \mathbf{f}_2^c we seek need to fulfill the static equilibrium conditions:

$$\underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{I} \\ \hat{\mathbf{r}}_1 & \hat{\mathbf{r}}_2 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \mathbf{f}_1^c \\ \mathbf{f}_2^c \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} -\mathbf{f}^g \\ \mathbf{0} \end{bmatrix}}_{\mathbf{b}}, \quad \mathbf{f}^g = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix}, \quad (3)$$

where \mathbf{r}_1 and \mathbf{r}_2 are the contact locations. Moreover, they need to satisfy the unilateral frictional constraints. To compute these desired ground reaction forces, we use the same formulation employed by our locomotion controller [10].

By approximating the friction cones with pyramids, the contact forces can be found by solving the following quadratic problem:

$$\begin{aligned} \text{minimize} \quad & (\mathbf{A}\mathbf{x} - \mathbf{b})^T \mathbf{S}(\mathbf{A}\mathbf{x} - \mathbf{b}) + \mathbf{x}^T \mathbf{W}\mathbf{x} \quad (4) \\ \text{s. t.} \quad & f_{n,i}^c \geq f_{n,\min}^c, \\ & \|\mathbf{f}_{t,i}^c\| \leq \mu f_{n,i}^c, \end{aligned} \quad (5)$$

where f_n^c is the component of the contact force along the surface normal and f_t^c the tangential one. The weighting matrix $\mathbf{S} = \text{diag}\{1, 1, 1, 20, 20, 10\}$ and $\mathbf{W} = \text{diag}\{10^{-5} \dots 10^{-5}\}$, friction coefficient $\mu = 0.8$ and mass $m = 27$ kg were used for this example.

The results of the computation are visualized in fig. 5. It can be seen that, due to the symmetry of the telescoping strut configuration, a set of valid contact forces that satisfy the static equilibrium objective (residual error is 0.001) can be found (fig. 5, left column). However, for the lever mechanism strategy, the different moment arms, the coupling between the net resulting force and the pitching and rolling moments, as

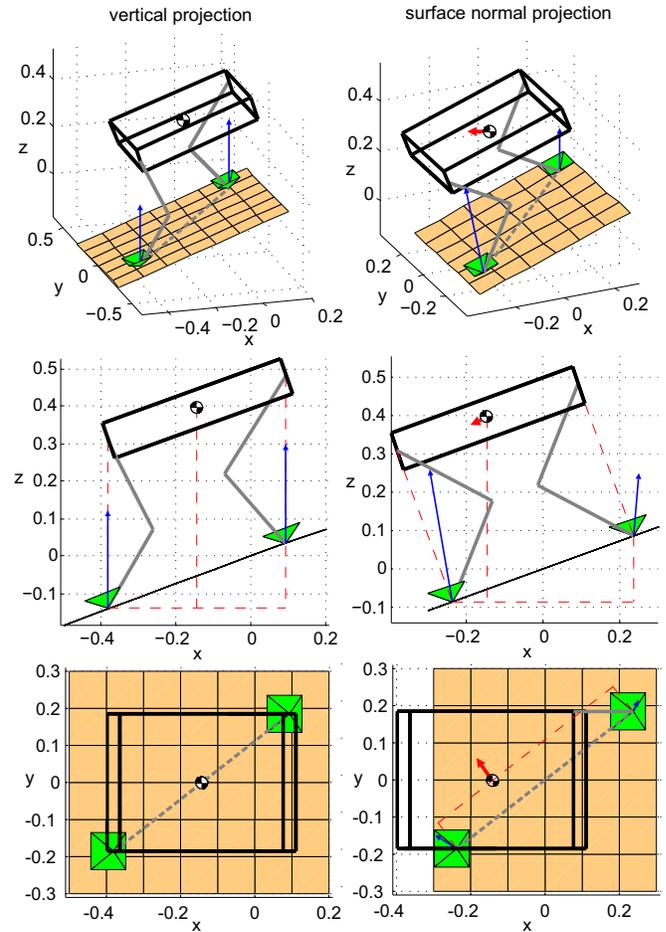


Fig. 5. The contact forces are presented as blue arrows for two different foot placement strategies. On the left side the contact points are vertically projected under the hips and a static equilibrium can be achieved while satisfying the frictional constraints indicated as green friction pyramids. On the right side the contact points are projected from the hips along the surface normal. No contact forces can be found that can fulfill a static equilibrium. The resulting error in net force of the optimization problem is indicated by the red arrow at the center of mass.

well as the friction cone constraints lead to a very significant error for the static equilibrium objective (residual error is 59.047). To reduce this error, the lateral location of the body relative to the feet would need to be further adapted by the control framework. This is certainly an interesting challenge, but it remains a direction of future investigation for our work. For the results demonstrated in this paper, we employ the telescoping strut strategy.

D. Adaptation of the Swing Leg Motion

A further consideration when locomoting on slopes is making appropriate adjustments to the trajectory of the swing feet. As illustrated in fig. 4c), if the trajectories of the feet are computed relative to a flat ground, the feet will touch-down early while walking uphill. The desired footholds will consequently not be reached, resulting in a loss of balance.

We compute the final target position of the swing feet using a combination of feedforward and feedback terms, as $\delta = \delta_{\text{ff}} + \delta_{\text{fb}}$. The feedforward term δ_{ff} is a function of the

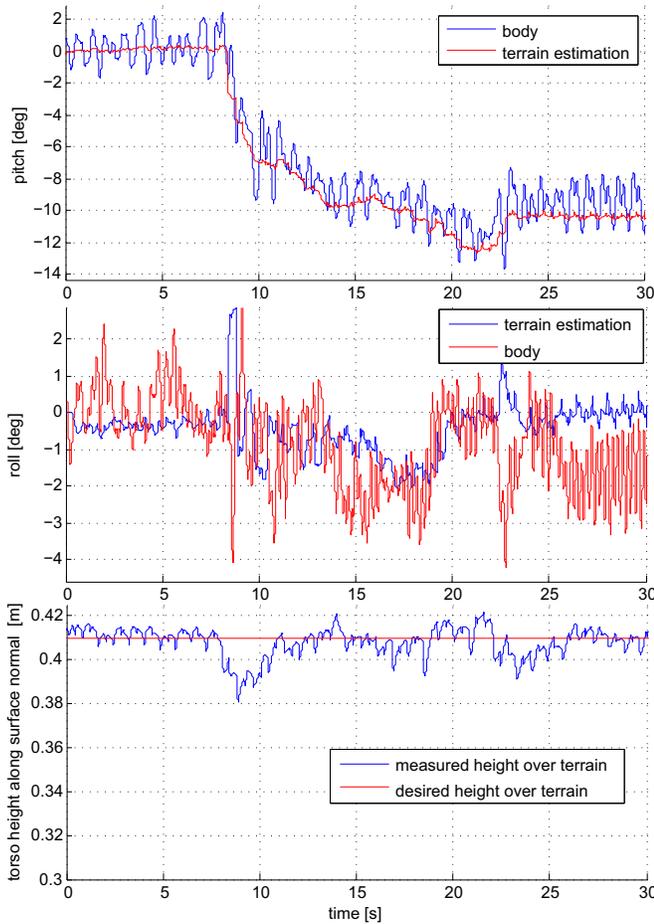


Fig. 6. The body pose adapts to changes in terrain height and inclination.

desired trotting speed, while the feedback term δ_{fb} further takes into consideration the robot's velocity. The target foot position is computed relative to the default configuration the leg is assumed to take during stance (i.e. telescoping strut vs lever mechanism). The target position of the foot during flight is computed by interpolating, as a function of the swing phase, the initial position of the foot at take-off and the final foot target, as described in detail in [10]. The desired height trajectory for the swing feet is adapted by adding the estimated height of the ground, as illustrated in fig. 4c).

E. Results

With the proposed modifications implemented in our control framework, Star1ETH was able to perform a dynamic trot on a variety of sloped terrains of various (10° , 16° , 21°). Several of our experiments can be seen in the accompanying video¹.

Figure. 6 shows measurements collected during an experiment where the robot starts trotting on flat terrain and then traverses a 10° -slope by moving forward. The robot was steered by following high-level velocity commands given by a joystick (forward, lateral and turning speed). The estimated pitch and roll angles of the terrain, the attitude of the torso,

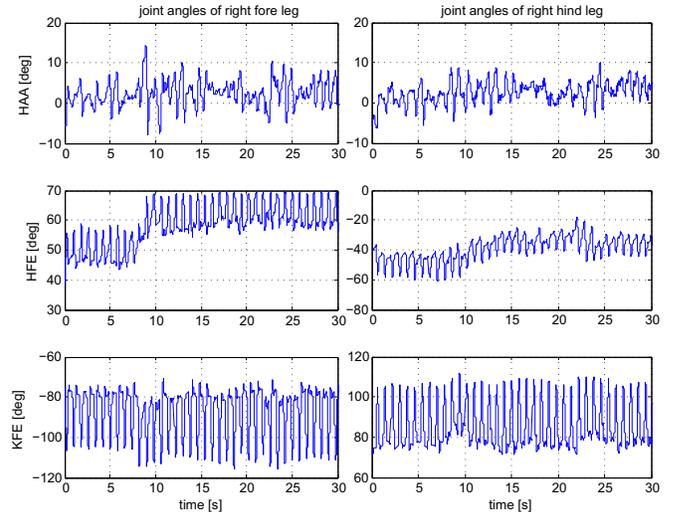


Fig. 7. Measured joint positions for right fore and hind legs for walking up a slope of 10° .

as well as the height of the robot above ground are shown in fig. 6.

The robot effectively adapts its pitch angle to match the terrain, while the tracking of the roll angle and the height are somewhat less accurate, $\pm 3^\circ$ and ± 2 cm respectively, which can also be observed while trotting on flat terrain. The measured joint angles for the right fore and hind legs are shown in fig. 7. The hip flexion/extension (HFE) and knee angle (KFE) provide a visual indication that the legs were tilted with respect to the robot while walking uphill in order to implement the telescoping strut strategy.

In the video, we also show the robot walking on a 21° -slope as shown in fig. 8 and turning on a 16° -slope, which verifies that the method works reliably when both pitch and roll slopes are present. This behavior is not possible on steeper slopes due to the stringent limits of the hip adduction/abduction joints. When moving forward or backward, Star1ETH was able to trot autonomously on slopes of up to 21° .

V. CONCLUSION

Our approach of estimating the terrain modeled as a plane and adapting the robot's configuration to it enabled dynamic trotting on slopes up to 21° (38%) for the quadruped robot Star1ETH. We improved and extended a model-based control framework for dynamic gaits, which showed robust locomotion on irregular but flat terrain.

The contribution of this work is an accurate method to extract slope angles and terrain heights using limited perception capabilities. The terrain is modeled as a simple plane, whose parameters are estimated from a history of footholds in a least-square sense. The proposed method is robust against drifts of measurements of the robot's pose and is applicable for dynamic gaits. The approach was successfully tested on a quadruped robot trotting on slopes with different angles and transitions from flat to the inclined terrain.

¹<http://youtu.be/NPuHwpxVUpg>

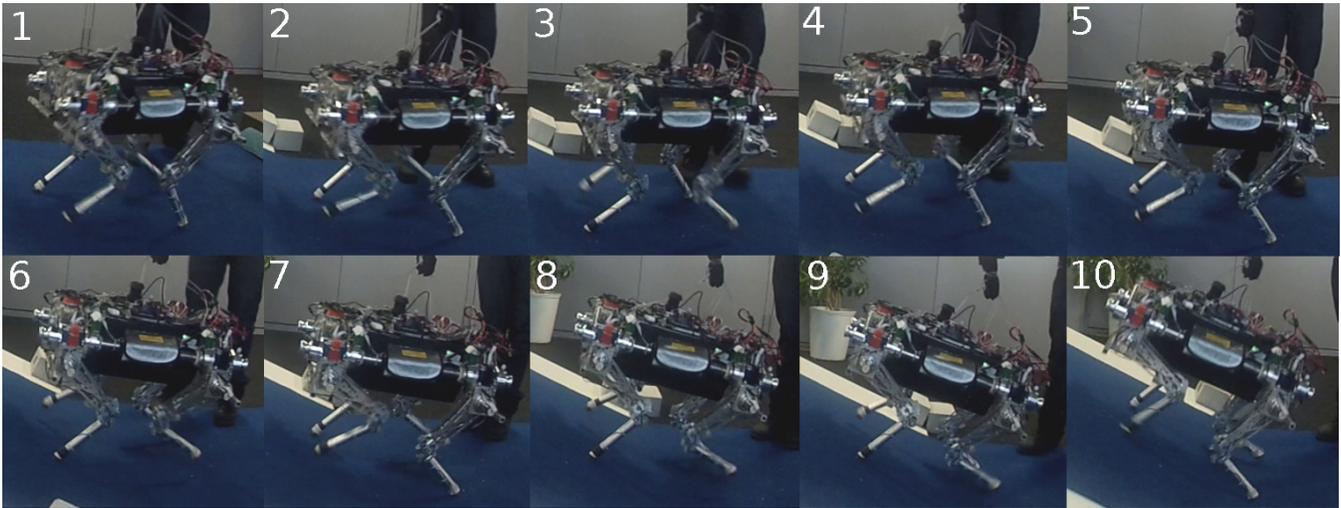


Fig. 8. The quadruped StarlETH trots fully power and computation autonomously on slopes. The sequence shows the robot trotting first on flat ground and then transiting on a 21°-slope in picture 6.

The second contribution is the control strategy based on the information of the estimated terrain model. Adapting the torso's attitude and height to the estimated slope helps to relax kinematic constraints. We have analyzed the influence of the foot placement with respect to the center of mass and concluded that the location of the foothold needs to be selected with care for a dynamic trot because only two support legs are present. Since the tangential forces need to contribute to the gravity compensation, the modulation of the contact forces is limited, and hence, the realizable motion. We applied a foot placement strategy that is called telescoping strut in biology, which places the feet vertically under the hips. This strategy assures that the distances between the contact points and the center of mass are equal in horizontal direction.

As future work, a second foot placement strategy, the so-called lever mechanism, should be examined, because the feet are placed in a way that the leg workspace is better oriented and thus kinematic limits are less of an issue for steep slopes as for the telescoping strut.

REFERENCES

- [1] D. V. Lee, "Effects of grade and mass distribution on the mechanics of trotting in dogs." *The Journal of experimental biology*, vol. 214, no. Pt 3, pp. 402–11, Feb. 2011.
- [2] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter, "Bigdog, the rough terrain quadruped robot," in *Proceedings of the 17th World Congress*, 2008, pp. 10 823–10 825.
- [3] B. Choi and S.-M. Song, "Fully automated obstacle-crossing gaits for walking machines," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18, no. 6, pp. 952–964, 1988.
- [4] J. Pan and J. Cheng, "Study of quadruped walking robot climbing and walking down slope," in *Proceedings IROS '91:IEEE/RSJ International Workshop on Intelligent Robots and Systems '91*, no. 9. IEEE, 1991, pp. 1531–1534.
- [5] Y. Fukuoka, H. Kimura, and a. H. Cohen, "Adaptive Dynamic Walking of a Quadruped Robot on Irregular Terrain Based on Biological Concepts," *The International Journal of Robotics Research*, vol. 22, no. 3-4, pp. 187–202, Mar. 2003.
- [6] R. Hodoshima, T. Doi, Y. Fukuda, S. Hirose, T. Okamoto, and J. Mori, "Development of TITAN XI: a quadruped walking robot to work on slopes," in *Intelligent Robots and Systems (IROS)*, vol. 1, 2004, pp. 792–797.
- [7] H. Kim, T. Kang, V. G. Loc, and H. R. Choi, "Gait Planning of Quadruped Walking and Climbing Robot for Locomotion in 3D Environment," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, vol. 2005. IEEE, 2005, pp. 2733–2738.
- [8] L. Righetti, J. Buchli, M. Mistry, M. Kalakrishnan, and S. Schaal, "Optimal distribution of contact forces with inverse-dynamics control," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 280–298, Jan. 2013.
- [9] A. Roennau, G. Heppner, M. Nowicki, J. M. Zoellner, and R. Dillmann, "Reactive Posture Behaviors for Stable Legged Locomotion over Steep Inclines and Large Obstacles," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 4888–4894.
- [10] C. Gehring, S. Coros, M. Hutter, M. Bloesch, M. A. Hoepflinger, and R. Siegwart, "Control of dynamic gaits for a quadrupedal robot," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, May 2013, pp. 3287–3292.
- [11] V. Barasuol, J. Buchli, C. Semini, M. Frigerio, E. R. De Pieri, and D. G. Caldwell, "A reactive controller framework for quadrupedal locomotion on challenging terrain," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, May 2013, pp. 2554–2561.
- [12] X. Zhang and H. Zheng, "Walking up and down hill with a biologically-inspired postural reflex in a quadrupedal robot," *Autonomous Robots*, vol. 25, no. 1-2, pp. 15–24, Dec. 2007.
- [13] M. Hutter, H. Sommer, C. Gehring, M. Hoepflinger, M. Bloesch, and R. Siegwart, "Quadrupedal locomotion using hierarchical operational space control," *The International Journal of Robotics Research*, vol. 33, no. 8, pp. 1047–1062, May 2014.
- [14] M. Hutter, C. Gehring, M. Hoepflinger, M. Bloesch, and R. Siegwart, "Towards combining Speed, Efficiency, Versatility and Robustness in an Autonomous Quadruped," *IEEE Transactions on Robotics (in Press)*, 2014.
- [15] C. Gehring, S. Coros, M. Hutter, M. Bloesch, P. Fankhauser, M. A. Hoepflinger, and R. Siegwart, "Towards Automatic Discovery of Agile Gaits for Quadrupedal Robots," in *IEEE International Conference on Robotics and Automation 2014*, 2014.
- [16] M. Bloesch, C. Gehring, P. Fankhauser, M. Hutter, M. A. Hoepflinger, and R. Siegwart, "State estimation for legged robots on unstable and slippery terrain," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Nov. 2013, pp. 6058–6064.