

# Hybrid Operational Space Control for Compliant Legged Systems

Marco Hutter, Mark A. Hoepflinger, Christian Gehring, Michael Bloesch, C. David Remy, Roland Siegwart  
Autonomous Systems Lab, ETH Zurich, Switzerland, mahutter@ethz.ch

**Abstract**—This paper introduces the concept of *hybrid operational space control*, a method that unifies kinematic tracking of individual joints with an inverse dynamics task space controller for the remainder of the robot. The proposed control strategy allows for a hierarchical task decomposition while simultaneously regulating the inner forces between the contact points. At the same time it improves fast tracking for compliant systems by means of appropriate low level position controllers. Introducing Star $ETH$ , a compliant quadrupedal robot, the applicability of the controller and the hardware is demonstrated in real-time simulations and hardware experiments. We perform static walking in challenging terrain and show how the controller can combine precise and fast position control with robust and compliant interaction with the environment.

## I. INTRODUCTION

Legged systems should interact softly with their environment to ensure robustness against disturbances such as terrain irregularities or slippage, to protect their hardware from damage through unexpected collisions, and to safely work hand-in-hand with human collaborators. In contrast thereto, precise foot or hand placement as well as exact trajectory execution for robust and versatile walking or reaching requires accurate and fast tracking with their feet and hands. While the first scenarios calls for *compliant, lightweight, and torque controlled* solutions, the latter are easier fulfilled on stiff systems with good *position control* performance. Research has made significant progress in both directions, yet the unification in one single device and control framework is still posing fundamental challenges.

Inspired by biology, novel design and actuation concepts found more and more their way into robotics. Manipulators such as the WAM arm [2] pushed the state of the art with respect to decreasing the inertia of moving segments by using sophisticated cable pulley systems that allow concentrating all actuators in the robot's base. A very promising approach to make systems additionally mechanically compliant, back-drivable, and torque controllable is the use of Series Elastic Actuators (SEA) [19]. In this context, 'compliant' means that elastic elements decouple the actuator from the joints (as done in the Meka robot arms [23]), therefore protect the actuator and gearboxes from unforeseen collisions, and make the systems inherently safe for interaction with humans. Pushing the state of the art with respect to compliant *legged systems*, we developed Star $ETH$  (Fig. 1, Section II, [11]) a quadruped robot that combines the advantages of the WAM and Meka arms (lightweight segments, compliant, torque controllable) and that additionally has the capacity to utilize its high-

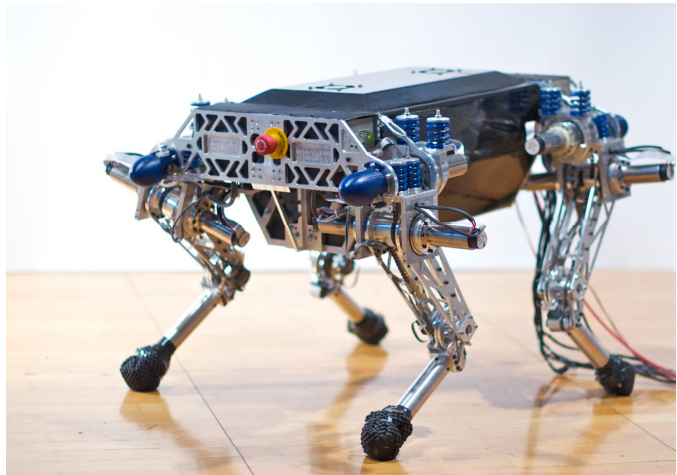


Fig. 1. The quadrupedal platform Star $ETH$  was developed to study **fast**, **efficient**, and **versatile** locomotion. It is actuated by high compliant series elastic actuators in all joints (photo François Pomerleau).

compliant elastic elements to periodically store energy [1] and thereby increase the efficiency of locomotion [4].

The transition from classical walking machines [5] to such a compliant system is accompanied by a shift from traditional and well elaborated position based control approaches to novel torque control based strategies. In doing so, the ongoing improvement in computational power allows the integration of increasingly complex model based control strategies, primarily based on inverse dynamics. As an example, Sentis et al. [25, 26] elaborated a sophisticated framework for humanoid robot control that included a floating base description with varying support constraints. It was recently released for public domain [18]. Along the lines of Khatib's seminal work on Operational Space Control (OSC) [12], it extends an inverse dynamics approach with a hierarchical task decomposition [27] in which multiple tasks are executed simultaneously in accordance with well-defined priorities. This results in a very powerful tool for controlling robots with a large amount of actuators.

A different approach for inverse dynamics based on orthogonal projection was presented by Mistry et. al. [15]. This method avoids the inversion of the inertia matrix, which makes it more robust against model uncertainty [17]. Recently, the same group summarized these different methods for inverse dynamics [21] showing that they are equivalent with respect to the minimization of different cost functions.

Using OSC techniques has the great advantage that even

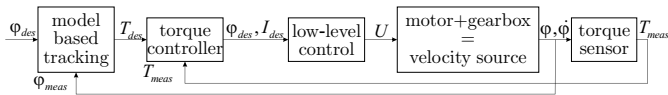


Fig. 2. A position tracking task in an inverse dynamics framework requires joint torque controllability. Since an electric actuator with a high gearbox reduction have to be considered rather as a velocity/position than a torque source, a cascaded control structures is applied which can lead to position control performance loss.

for highly complex robots the dynamics get controllable in a very intuitive way. By defining the kinematic behavior of a set of distinguished task points and by choosing a task specific stiffness and damping behavior, position control tasks (such as following a certain foot-point trajectory or moving the Center of Gravity (CoG)) are mapped into task space accelerations and subsequently (via inverse dynamics methods) into joint torques. However, such a control framework assumes perfect torque actuators on joint level, which are generally not available in real robotic devices. Common electrical actuators in particular require high gearbox reductions and are consequently not backdrivable. For practical purposes, they must be considered as a velocity rather than a torque source, and only the integration of additional torque sensors at the gearbox output and a cascaded low-level regulator allows to make the robot fully torque-controllable (Fig. 2). Since the achievable controller fidelity (in terms of bandwidth and feedback gains) is lowered within every additional loop of the cascaded structure, this setup can greatly limit performance in practice. For a mechanically stiff system (such as for example the DLR robot arm [6]) or ‘soft’ tasks (such as interacting with the environment) this is not too much of a problem, but for mechanically compliant systems (such as the Meka Arm or similar robots with a torque bandwidth of below 5Hz [20]) and precise positioning tasks (such as foot placement) this can quickly become critical. Consequently, for purely kinematic tasks, a low level joint position controller that compensates for the known series elasticity (e.g., a LQG (Linear-Quadratic-Gaussian) structure presented in [10]) will always perform better than an inverse dynamics framework in a cascaded setup as depicted in Fig. 2.

We accordingly consider locomotion of compliant legged systems as a hybrid control problem. Parts of the robot that are conducting high performance tracking tasks (in particular swing leg control) will be locally position controlled, while the remainder of the robot will compliantly interact with the environment in an inverse dynamics framework to ensure a robust behavior of the system. In this paper we present an approach for hybrid OSC that combines these two requirements: Similar to [27], all dynamic tasks are brought into a prioritized task space control structure while certain joints are position controlled based on traditional inverse kinematics control methods. This framework measures and estimates the influence of the position controlled joints on the torque controlled parts such that we can compensate for these effects and achieve exactly the same task space behavior as with a purely torque controlled system, while at the same time tracking performance can be improved. Several particular

cases of legged locomotion or robot-human interaction fit perfectly into this scenario. Yet, in this paper we focus on quadrupedal walking using StarLETH in simulations and for experimental validation.

## II. HARDWARE DEVELOPMENT

StarLETH (Springy Tetrapod with Articulated Robotic Legs, Fig. 1) is a quadruped robot that was built at the Autonomous Systems Lab to study **fast**, **efficient**, and **versatile** locomotion. The system has a size of  $710 \times 640 \times 580$  mm with segment lengths of 200 mm, a total weight of 23 kg (2.4 kg per leg, 13.2 kg for the main body), and 12 actuated degrees of freedom, all driven by high compliant SEAs. The leg design is based on ScarLETH (Series Compliant Articulated Robotic Leg), a prototype leg that served as a test bench for single legged planar running, to evaluate hardware performance, and to design low level controllers [9, 10]. The main focus in terms of design was put on the high compliant SEAs: The gearbox output of all three degrees of freedom (hip and knee flexion/extension, as well as for hip abduction/adduction) are connected through cable pulley systems and chain drives to antagonistically pre-loaded compression springs. This allowed attaching all actuators closely to the main body and hence minimizing the mass and inertia of the moving segments.

Despite the loss in control bandwidth due to the high mechanical compliance, SEAs combine multiple advantages for legged locomotion. In addition to the decoupling mechanism of the elasticity that makes the system robust against impacts and hence well-suited for highly dynamic maneuvers, the spring substantially contributes to the passive dynamics of the system and allows for passive energy storage during stance. In single legged hopping, the amount of energy that is recovered amounts to approximately 70% while only about 30% of the energy-fluctuations are actually coming from the actuators to compensate for damping and impact losses [9]. Since the springs show a nearly perfect linear characteristic with minimal mechanical damping, accurate torque control can be achieved through deflection control [9]. In addition thereto it could be shown that using an appropriate LQG control structure ensures fast joint tracking [10] with a position control bandwidth (9 Hz) approximately equal to the torque control bandwidth ( $\approx 11$  Hz)[9]. It actively suppresses undesired oscillations that occur due to the passive dynamics of spring - providing highly better performance than if one would use a cascaded structure as in Fig. 2.

## III. INVERSE DYNAMICS

The equation of motion (EoM) for a walking machine can be stated in the form

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g} + \mathbf{J}_s^T \mathbf{F}_s = \mathbf{S}^T \boldsymbol{\tau}, \quad (1)$$

with the mass matrix  $\mathbf{M}(\mathbf{q})$ , the coriolis and centrifugal contribution  $\mathbf{b}(\mathbf{q}, \dot{\mathbf{q}})$ , the gravitational component  $\mathbf{g}(\mathbf{q})$ , the ground contact force  $\mathbf{F}_s$ , its corresponding Jacobian  $\mathbf{J}_s(\mathbf{q})$  as a function of the generalized coordinates  $\mathbf{q} = [\mathbf{q}_b; \mathbf{q}_r] \in \mathbb{R}^n$ , and the actuator torque  $\boldsymbol{\tau} \in \mathbb{R}^{n-6}$ . The selection matrix

$\mathbf{S} = [\mathbf{0}_{(n-6) \times 6}, \mathbf{I}_{(n-6)}]$  separates the actuated joint coordinates  $\mathbf{q}_r \in \mathbb{R}^{n-6}$  from the floating base coordinates  $\mathbf{q}_b \in \mathbb{R}^6$ .

The ground contact force  $\mathbf{F}_s$  is the constraining force that appears due to the contact condition

$$\dot{\mathbf{r}}_s = \mathbf{J}_s \dot{\mathbf{q}} = \mathbf{0}, \quad \ddot{\mathbf{r}}_s = \mathbf{J}_s \ddot{\mathbf{q}} + \dot{\mathbf{J}}_s \dot{\mathbf{q}} = \mathbf{0}, \quad (2)$$

whereby  $\mathbf{r}_s$  represents a stacked array of all active contact points. In order to eliminate the contact force for inverse dynamics, there exist different approaches which can be regarded as a support null space projection of the dynamics:

$$\mathbf{P}(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g}) = \mathbf{P}\mathbf{S}^T \boldsymbol{\tau} \quad (3)$$

$$\mathbf{P}\mathbf{J}_s^T = \mathbf{0} \quad \forall \mathbf{q} \quad (4)$$

As it was recently shown by Righetty et al. [21], OSC techniques for floating base systems [27] can be brought to this inverse dynamics description with

$$\mathbf{P}_{OSC} = \mathbf{S}\mathbf{N}_s, \quad (5)$$

whereby  $\mathbf{N}_s = \mathbf{M}^{-1} \left( \mathbf{I} - \mathbf{J}_s^T (\mathbf{J}_s \mathbf{M}^{-1} \mathbf{J}_s^T)^{-1} \mathbf{J}_s \mathbf{M}^{-1} \right)$  represents the dynamically consistent null space of the supporting contacts.

Similar to that, the QR decomposition [15] of the support Jacobian  $\mathbf{J}_s^T = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$  with  $\mathbf{Q}^T = \mathbf{Q}^{-1}$  and the upper-right triangular matrix  $\mathbf{R}$  in combination with a selection matrix  $\mathbf{S}_u = [\mathbf{0} \quad \mathbf{I}]$  can be brought to this form with

$$\mathbf{P}_{QR} = \mathbf{S}_u \mathbf{Q}^T. \quad (6)$$

The same works also with a direct kinematic null space projection of the support Jacobian  $\mathbf{J}_s$  that can be weighted e.g. with the mass matrix

$$\mathbf{P}_{dir} = \mathbf{I} - \mathbf{J}_s^T (\mathbf{J}_s \mathbf{M}^{-1} \mathbf{J}_s^T)^{-1} \mathbf{J}_s \mathbf{M}^{-1}. \quad (7)$$

The following sections will be independent of the choice of this null space projection. Given a desired acceleration, all these methods allow to directly calculate the required joint torques via a pseudo inverse

$$\boldsymbol{\tau}_m = (\mathbf{P}\mathbf{S}^T)^+ \mathbf{P}(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g}), \quad (8)$$

whereby the subscript  $m$  indicates that the torques induce motion (compare to Section VI). The pseudo-inverse of (8) minimizes the cost function  $\boldsymbol{\tau}_m^T \boldsymbol{\tau}_m$ . Changing the pseudo-inverse to the more generalized form  $(\mathbf{P}\mathbf{S}^T)^{\bar{\mathbf{W}}} = \mathbf{W}^{-1} \mathbf{S} \mathbf{P}^T (\mathbf{P}\mathbf{S}^T \mathbf{W}^{-1} \mathbf{S} \mathbf{P}^T)^+ [21]$  allows to change the cost minimization of the inversion to  $\boldsymbol{\tau}_m^T \mathbf{W} \boldsymbol{\tau}_m$  which can account for different segment masses or the available actuator power.

#### A. Task prioritization

In contrast to the hierarchical task decomposition of [27], we use a purely kinematic task prioritization approach that has the benefit of avoiding any mass matrix inversions.

In general, the total generalized acceleration  $\ddot{\mathbf{q}}$  can be written as the sum of task-induced accelerations  $\ddot{\mathbf{q}}_j$  projected into the null space  $\mathbf{N}_{j-1}$  of all higher prioritized tasks

$$\ddot{\mathbf{q}} = \sum_{j=1}^n \mathbf{N}_{j-1} \ddot{\mathbf{q}}_j, \quad (9)$$

with  $\mathbf{N}_0$  being the unitary matrix. The relation between joint space  $\ddot{\mathbf{q}}$  and task space  $\ddot{\mathbf{r}}$  is given through

$$\ddot{\mathbf{r}}_i = \mathbf{J}_i \ddot{\mathbf{q}} + \dot{\mathbf{J}}_i \dot{\mathbf{q}}. \quad (10)$$

Combining (9) with (10) allows to solve for  $\ddot{\mathbf{q}}_i$ :

$$\mathbf{J}_i \ddot{\mathbf{q}} = \mathbf{J}_i \sum_{j=1}^i \mathbf{N}_{j-1} \ddot{\mathbf{q}}_j = \ddot{\mathbf{r}}_i - \dot{\mathbf{J}}_i \dot{\mathbf{q}} \quad (11)$$

$$\ddot{\mathbf{q}}_i = (\mathbf{J}_i \mathbf{N}_{i-1})^+ \left( \ddot{\mathbf{r}}_i - \dot{\mathbf{J}}_i \dot{\mathbf{q}} - \mathbf{J}_i \sum_{j=1}^{i-1} \mathbf{N}_{j-1} \ddot{\mathbf{q}}_j \right) \quad (12)$$

To ensure that the lower prioritized tasks do not influence higher prioritized tasks, it has to hold that

$$\mathbf{J}_j \mathbf{N}_i = \mathbf{0} \quad \forall j \leq i. \quad (13)$$

This is fulfilled if  $\mathbf{N}_i$  represents the null space of the stacked Jacobian matrix of all higher or equal prioritized tasks:

$$\mathbf{N}_i = \mathcal{N}([\mathbf{J}_1; \dots; \mathbf{J}_i]) \quad (14)$$

In a very general way, the null space  $\mathcal{N}$  can be found through singular value decomposition  $[\mathbf{S}, \mathbf{V}, \mathbf{D}] = \text{svd}(\mathbf{J})$  which shows better numerical stability than a kernel computation using an LU-decomposition. Since  $\mathbf{V}$  defines an orthonormal basis, the null space is defined through the basis vectors that have a zero singular value  $\mathcal{N} = \mathbf{V}(:, \mathbf{S}(i, i) == 0)$ .

The complete procedure can be stated in the following recursive algorithm for task prioritization:

---

#### Algorithm 1 Recursive task prioritization for accelerations

---

$n$  = Number of Tasks

$\ddot{\mathbf{q}}_t = \mathbf{0}$  % total acceleration

$\mathbf{N}_0 = \mathbf{I}$  % initial null space

**for**  $i = 1 \rightarrow n$  **do**

$\mathbf{N}_i = \mathcal{N}([\mathbf{J}_1; \dots; \mathbf{J}_i])$

$\ddot{\mathbf{q}}_i = (\mathbf{J}_i \mathbf{N}_{i-1})^+ \left( \ddot{\mathbf{r}}_i - \dot{\mathbf{J}}_i \dot{\mathbf{q}} - \mathbf{J}_i \ddot{\mathbf{q}}_t \right)$

$\ddot{\mathbf{q}}_t \leftarrow \ddot{\mathbf{q}}_t + \mathbf{N}_{i-1} \ddot{\mathbf{q}}_i$

**end for**

---

*Note:* This Section holds for floating as well as for fixed base systems. In the case of floating base systems, the highest priority task includes mandatorily all support acceleration constraints (2). This is equivalent to the support null space description in the task hierarchy of [25]. Otherwise, the joint accelerations are not consistent with the support constraints.

#### IV. ANALYTICAL GLOBAL KINEMATICS AND DYNAMICS

Model based control requires fast and simple methods to get continuously updated global kinematics and dynamics of the robotic system. In contrast to most approaches where this is provided by a simulation environment, we developed a tool [8] to get the global kinematics and dynamics (1) in an analytical representation. Using the MATLAB Symbolic Toolbox<sup>1</sup> a relative kinematic tree of the robot and locally defined force elements acting in the joints are translated to a global kinematic description using Euler rotations. The system dynamics, respectively the EoM in the form of (1) are established using projected Newton-Euler equations:

$$\mathbf{M} = \sum_{i=1}^N \mathbf{J}_{S_i}^T m_i \mathbf{J}_{S_i} + \mathbf{J}_{R_i}^T \boldsymbol{\theta}_{S_i} \mathbf{J}_{R_i} \quad (15)$$

$$\mathbf{b} = \sum_{i=1}^N \mathbf{J}_{S_i}^T m_i \ddot{\mathbf{J}}_{S_i} \dot{\mathbf{q}} + \mathbf{J}_{R_i}^T \boldsymbol{\theta}_{S_i} \ddot{\mathbf{J}}_{R_i} \dot{\mathbf{q}} + \boldsymbol{\Omega}_i \times \boldsymbol{\theta}_{S_i} \boldsymbol{\Omega}_i \quad (16)$$

$$\mathbf{g} = \sum_{i=1}^N -\mathbf{J}_{S_i}^T \mathbf{F}_{S_i}^g \quad (17)$$

with the translational Jacobians  $\mathbf{J}_{S_i} = \frac{\partial \mathbf{r}_{S_i}}{\partial \mathbf{q}}$  evaluated at the CoG of all segments, the rotational Jacobians  $\mathbf{J}_{R_i} = \frac{\partial \boldsymbol{\Omega}_i}{\partial \dot{\mathbf{q}}}$ , the rotational speed  $\boldsymbol{\Omega}_i$ , mass  $m_i$ , the local body inertia matrix  $\boldsymbol{\theta}_{S_i}$ , and the gravity force vector  $\mathbf{F}_{S_i}^g$ . The code is open source<sup>2</sup> and simple to adapt for every individual or controller specific purpose. Using an analytical representation allows very fast controller routines that are independent of a particular simulation environment.

#### V. HYBRID OPERATIONAL SPACE CONTROL

For high performance joint position or end-effector tracking, it is often beneficial to rely on inverse kinematics in combination with fast low-level position controllers for certain joints, while major parts of the system undergo a hierarchical task decomposition with inverse dynamics as presented in Section III. A classic example for such a scenario is quadrupedal walking as depicted in Fig. 3. The legs that are in ground contact (blue solid lines) remain torque controlled based on inverse dynamics to move/stabilize the main body and to modify contact forces, while fast and precise tracking of the swing foot (red dotted line) is considered as an inverse kinematics problem. There are no direct means of regulating the acceleration of these position controlled joints  $\ddot{\mathbf{q}}_k = \mathbf{S}_k \dot{\mathbf{q}}$ . For lightweight segments and slow maneuvers, coupling effects of the swing leg motion on the system dynamics are negligible (Section V-A). For faster maneuvers, it is required to compensate for these effects through measuring (Section V-B), estimating (Section V-C), or predicting (Section V-D) the corresponding joint acceleration  $\ddot{\mathbf{q}}_k$  and to consider it as an additional task in Algorithm 1.

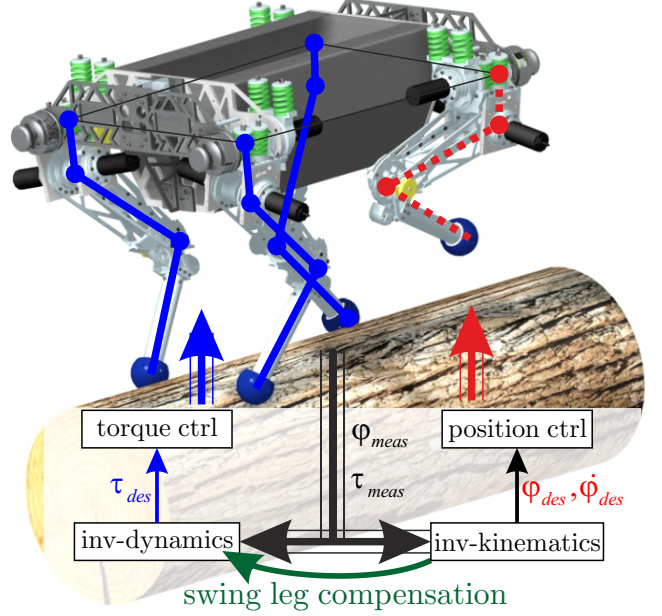


Fig. 3. Walking on a tree stem requires precise foot tracking, CoG stabilization, and internal force regulation to avoid slipping. For compliant systems this is considered as a combination of kinematic position tracking (swing leg) and inverse dynamics control with swing leg compensation.

##### A. No compensation

Since in most cases only lightweight end-effectors require accurate position tracking (walking, grasping, etc.), the corresponding influence on the remainder of the body can be neglected ( $\hat{\mathbf{q}}_k = \mathbf{0}$ ). This method is appropriate for slow maneuvers.

##### B. Acceleration measurement

Using a joint encoder, the acceleration can be measured through double differentiation of the position signal ( $\ddot{\mathbf{q}}_k$ ). While being theoretically the best solution, this method is inapplicable in practice due to sensor noise.

##### C. Acceleration estimation

Torque sensing in SEAs is mostly done through a deflection (position) measurement of the series spring. Since this large compliance acts as a mechanical low-pass filter, torque signals  $\tilde{\boldsymbol{\tau}}$  are in general very accurate and clean [10]. Hence, the joint acceleration can be estimated as

$$\hat{\ddot{\mathbf{q}}}_k = \mathbf{S}_k (\mathbf{P}\mathbf{M})^+ \mathbf{P} (\mathbf{S}^T \tilde{\boldsymbol{\tau}} - \mathbf{b} - \mathbf{g}). \quad (18)$$

##### D. Acceleration prediction

In applications as presented here, only end-effector joints will be position controlled. Therefore, all other objectives in the dynamic task prioritization defined in Section III-A are independent of the position controlled joints ( $\mathbf{S}_k \mathbf{J}_i^T = \mathbf{0}$ ). This relation allows to encapsulate the EoM for the position controlled joints through pre-multiplying (1) with  $\mathbf{S}_k$  (selecting only dynamics for position controlled joints) and through

<sup>1</sup>www.mathworks.com/products/symbolic

<sup>2</sup>www.leggedrobotics.ethz.ch/software/proneu

inserting the separation  $\ddot{\mathbf{q}} = \mathbf{S}_d \ddot{\mathbf{q}}_d + \mathbf{S}_k \ddot{\mathbf{q}}_k$  of torque ( $\mathbf{S}_d$ ) and position ( $\mathbf{S}_k$ ) controlled joints:

$$\hat{\ddot{\mathbf{q}}}_k = (\mathbf{S}_k \mathbf{M} \mathbf{S}_k^T)^{-1} (\tilde{\boldsymbol{\tau}}_k - \mathbf{S}_k \mathbf{M} \mathbf{S}_d^T \ddot{\mathbf{q}}_d - \mathbf{S}_k (\mathbf{b} + \mathbf{g})) \quad (19)$$

In contrast to the estimation approach, the prediction of the acceleration  $\hat{\ddot{\mathbf{q}}}_k$  is purely based on swing leg torque measurements  $\tilde{\boldsymbol{\tau}}_k$  in combination with the desired joint acceleration  $\ddot{\mathbf{q}}_d$  given by the task prioritization.

## VI. GROUND CONTACT FORCES

In addition to the prioritized motion control of the task points, the presented framework allows advanced manipulation of the contact forces of multi-contact systems [22] via the null space of the pseudo-inversion (8), which represents the internal forces  $\mathbf{F}_{s0}$  that do not influence the task motion:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_m + \mathcal{N}(\mathbf{P} \mathbf{S}^T) \boldsymbol{\tau}_0 = \boldsymbol{\tau}_m + \mathbf{N}_P \boldsymbol{\tau}_0 \quad (20)$$

$\boldsymbol{\tau}_0$  allows to augment the total contact forces by

$$\mathbf{F}_{s0} = (\mathbf{J}_s^T)^+ \mathbf{S}^T \mathbf{N}_P \boldsymbol{\tau}_0 = \mathbf{A} \boldsymbol{\tau}_0, \quad (21)$$

where  $\mathbf{A}$  spans the subspace of  $\mathbf{F}_{s0}$  and  $\text{rank}(\mathbf{A})$  gives the number of internal directions. This allows modifying the total contact forces to

$$\mathbf{F}_s = \mathbf{F}_{sm} + \mathbf{F}_{s0}, \quad (22)$$

with the motion implied ground contact forces  $\mathbf{F}_{sm}$  given through

$$\mathbf{F}_{sm} = \mathbf{J}_s^+ (\mathbf{S}^T \boldsymbol{\tau}_m - (\mathbf{M} \ddot{\mathbf{q}} + \mathbf{g} + \mathbf{b})). \quad (23)$$

There are several interesting and relevant cases where the contact forces can be adapted to minimize certain cost criterion:

$$\underset{\boldsymbol{\tau}_0}{\text{minimize}} \quad f(\mathbf{F}_{sm} + \mathbf{A} \boldsymbol{\tau}_0). \quad (24)$$

### A. No internal forces

The system has no internal forces which is equivalent to a minimal total ground contact forces:

$$\underset{\boldsymbol{\tau}_0}{\text{minimize}} \quad \|(\mathbf{F}_{sm} + \mathbf{A} \boldsymbol{\tau}_0)\|_2 \quad (25)$$

This is a very common approach in minimizing slippage when walking on flat ground [28]. Given that the pseudo-inverse solves the least square error problem, the solution is

$$\boldsymbol{\tau}_0 = -\mathbf{A}^+ \mathbf{F}_{sm}. \quad (26)$$

### B. Preferred normal force directions

For walking in rough terrain it becomes very important to align all contact forces with the local surface normal directions  $\mathbf{n}_i \in \mathbb{R}^{3 \times 1}$ , respectively to minimize the forces in the corresponding tangential planes  $\mathbf{t}_i \in \mathbb{R}^{3 \times 2}$ . Using a tangential plane selection matrix  $\mathbf{D}$  and a geometric weighting factor  $\alpha_i$  that allows to account for the stance-geometry, the minimization problem (25) can be written as

$$\underset{\boldsymbol{\tau}_0}{\text{minimize}} \quad \|\mathbf{D}^T (\mathbf{F}_{sm} + \mathbf{A} \boldsymbol{\tau}_0)\|_2 \quad (27)$$

with the tangential projection matrix

$$\mathbf{D}^T = \text{blockdiagonal}(\alpha_i \mathbf{t}_i^T). \quad (28)$$

The corresponding solution is:

$$\boldsymbol{\tau}_0 = -(\mathbf{D}^T \mathbf{A})^+ \mathbf{D}^T \mathbf{F}_{sm}. \quad (29)$$

### C. Constrained minimization problem

Ideally, one would like to adjust the ground contact forces in a way that the maximally required friction coefficient  $\mu_{max} = \left\| \frac{\mathbf{F}_{st}^t}{\mathbf{F}_{st}^n} \right\|_\infty$  is kept small, whereby superscript  $n$  and  $t$  indicate the normal respectively tangential force directions of the grounded legs  $i \in \{1 \dots 4\}$ . This can be either seen as a minimization of the friction coefficients under torque constraints:

$$\begin{aligned} & \underset{\boldsymbol{\tau}_0}{\text{minimize}} \quad \mu_{max}(\boldsymbol{\tau}_0) \\ & \text{subject to} \quad \boldsymbol{\tau}_{min} < \boldsymbol{\tau} = \boldsymbol{\tau}_d + \mathbf{N}_P \boldsymbol{\tau}_0 < \boldsymbol{\tau}_{max}, \end{aligned} \quad (30)$$

or as a minimization of the joint torques while a certain safety against slipping has to be ensured:

$$\begin{aligned} & \underset{\boldsymbol{\tau}_0}{\text{minimize}} \quad (\boldsymbol{\tau}_m + \mathbf{N}_P \boldsymbol{\tau}_0)^T \mathbf{W} (\boldsymbol{\tau}_m + \mathbf{N}_P \boldsymbol{\tau}_0) \\ & \text{subject to} \quad \mu_{max}(\boldsymbol{\tau}_0) < \mu_{safety} \end{aligned} \quad (31)$$

### D. Selective force manipulation

Let  $\mathbf{S}_{leg}$  be a selection matrix of certain contact forces such that  $\mathbf{F}_{des} = \mathbf{S}_{leg} \mathbf{F}_s$  can be freely chosen. In this case,  $\boldsymbol{\tau}_0$  is given through

$$\boldsymbol{\tau}_0 = (\mathbf{S}_{leg} \mathbf{A})^+ (\mathbf{F}_{des} - \mathbf{S}_{leg} \mathbf{F}_{sm}). \quad (32)$$

As long as the matrix  $\mathbf{S}_{leg} \mathbf{A}$  has full rank, the desired forces are feasible, otherwise this results in the least square minimization of  $\|\mathbf{S}_{leg} \mathbf{F}_s - \mathbf{F}_{des}\|_2$ . This approach can be used to vary the force distribution between the different legs to achieve both smooth contact force and joint torque profiles. This is important for quadrupedal walking (Section VII-D2) when support changes.

## VII. SIMULATION

To demonstrate the performance of this control framework, we accomplish a static walking test on a tree stem (Fig. 3). This requires *i*) precise control of swing foot trajectories which is considered as a kinematic control task, *ii*) CoG control to ensure stability based on a hierarchical OSC implementation, and *iii*) internal force regulation to avoid slipping on the tree stem. The simulations are conducted both in an idealized MATLAB simulation with hard point contacts (impact at landing) as well as in a real-time control and simulation environment (Section VIII-A) that closely fits the actual system.

### A. Foothold and CoG trajectories

The robot periodically executes a constant footfall sequence [14] left-hind (LH), left-front (LF), right-hind (RH), right-front (RF). The way-points for the main body are planned through intersecting subsequent support triangles with a predefined safety margin for three steps ahead such that the body motion is minimized. Given the support center before stepping  $\mathbf{r}_c^- =$

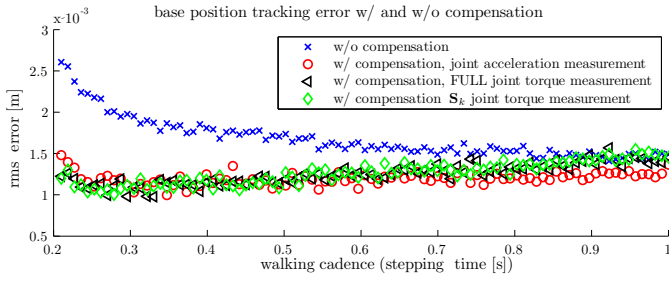


Fig. 4. Despite the low leg inertia, fast kinematic tracking of the swing leg introduces increased errors on the main body tracking. Through measuring joint accelerations or torques, these effects can be compensated.

$mean(\sum \mathbf{r}_{F_i}^-)$  (index  $-$  indicates before step), the next contact position  $\mathbf{r}_{F_i}^+$  of leg  $i$  is given through  $\mathbf{r}_{F_i}^+ = \mathbf{r}_c^- + \mathbf{r}_s + \mathbf{r}_{O_i}$  with the step vector  $\mathbf{r}_s$  and the constant offset value  $\mathbf{r}_{O_i}$  that describes the leg configuration with respect to the contact center in a predefined home position. As shown in [13] this converges to a symmetric pattern that staggers the footsteps in an optimal manner. The swing leg trajectory is defined through  $\mathbf{r}_{F_i}(t) = \mathbf{r}_{F_i}^- + (\mathbf{r}_{F_i}^+ - \mathbf{r}_{F_i}^-) \left( \frac{1}{2} - \frac{1}{2} \cos\left(\pi \frac{t}{t_{step}}\right) \right)$  whereby ground clearance is ensured through a superimposed vertical motion  $h_{F_i}(t) = \frac{h}{2} - h \cos\left(\pi \frac{t}{t_{step}}\right)$ .

### B. Task hierarchy with kinematic foot tracking

To ensure stability, the highest priority task is the CoG or main body base ( $\mathbf{r}_b, \mathbf{J}_b$ ) control. Additionally to that, the swing leg contact point has to follow the predefined step trajectory ( $\mathbf{r}_f, \mathbf{J}_f$ ) which is considered as a position control task, while the base orientation should be stabilized ( $\mathbf{r}_o, \mathbf{J}_o$ ). The last task (although in this special case not necessary) in the hierarchy is always a null space joint damping or posture control ( $\mathbf{r}_j, \mathbf{J}_j$ ). The desired task accelerations for the base position as well as the base orientation are determined based on a PD control law  $\ddot{\mathbf{r}}_t = k_t(\mathbf{r}_{des} - \mathbf{r}_t) + d_t(\dot{\mathbf{r}}_{des} - \dot{\mathbf{r}}_t) + \ddot{\mathbf{r}}_{des}$  with relatively low gains to achieve a robust and compliant behavior.

### C. Compensation for swing leg motion

To demonstrate the applicability of the hybrid setup (Section V) and to demonstrate the need for swing leg compensation, a series of flat ground walking simulations in the real-time environment was conducted over a fixed distance (2 m). Thereby, only the swing duration of every step was modulated in the range of 0.2 s to 1.0 s and the average absolute tracking error of the base position during swing phase was recorded. The results depicted in Fig. 4 are as expected: While for slow tracking the error of assuming zero joint acceleration (Section V-A, blue crosses) is negligible due to its very low inertia, it becomes crucial as the swing time is shortened. All other methods (Section V-B - V-D) perform equally good and (nearly) independent of the swing leg duration.

### D. Contact force optimization

Optimizing the contact forces without influencing the task tracking performance (Section VII-A) has two main purposes. First, the forces should be optimally aligned with the surface normals, such that the risk of slipping and losing ground

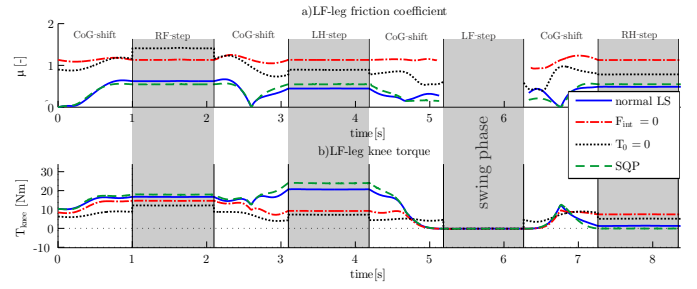


Fig. 5. a) Required friction coefficient of different force distribution methods in the tree stem experiment: the best results were achieved with the normal least-square solution (blue-solid, Section VI-B) and the constraint minimization problem (green-dashed, Section VI-C). Both the joint torque (black-dotted,  $\tau_0 = \mathbf{0}$ ) and the contact force minimization (red-dashed-dotted, Section VI-A) show unsatisfying results. b) Using (33), the contact force and joint torques are both smooth without any discontinuities.

contact can be minimized (Section VII-D1). Second, the contact force distribution has to be changed to get smooth transitions before and after lift-off and landing of each leg (Section VII-D2).

1) *Force alignment:* In Section VI we presented three different methods to change the ground contact force distribution. The least square solutions to minimize the total force (26) and to minimize the tangential forces (29) can be directly implemented using a pseudo-inverse. The constraint minimization (30) is implemented using a SQP (sequential quadratic programming) solver which is initialized using the least-square solution (29). These last two methods both require knowledge about the ground contact surface normal directions. Various studies have shown that relatively simple methods allow reliable foot surface shape detection based on haptic [7] or on visual feedback [16]. In simulations of a static gait executed on a tree stem, the required friction coefficient can be significantly lowered by taking the local surface normal directions into account (Fig. 5a). The constraint optimization (green dashed) distributes  $\mu$  equally on all legs. The normal direction weighted least square solution (blue solid) performs quite as good with significantly less computational effort.

2) *Load distribution:* During the CoG shifting phase from support triangle  $i$  to  $j$  (where  $i$  and  $j$  stand for two successive 3-sets of the 4 legs), the contact force is linearly modulated with the relative projected distance  $x \in [0..1]$  given through  $x = \frac{(\mathbf{r}_b - \mathbf{r}_0) \cdot \mathbf{n}_{01}}{(\mathbf{r}_1 - \mathbf{r}_0) \cdot \mathbf{n}_{01}}$  with the unit direction vector  $\mathbf{n}_{01} = \frac{\mathbf{r}_1 - \mathbf{r}_0}{\|\mathbf{r}_1 - \mathbf{r}_0\|}$ . This results in a desired ground contact force

$$\mathbf{F}^{desired} = \begin{cases} 2x\mathbf{F}_s^{\{4\}} + (1-2x)\mathbf{F}_s^{\{i\}} & x \leq 0.5 \\ 2(1-x)\mathbf{F}_s^{\{4\}} + (2x-1)\mathbf{F}_s^{\{j\}} & x > 0.5. \end{cases} \quad (33)$$

Without this contact force optimization ( $\tau_0 = \mathbf{0}$ , black dotted line), the contact force (Fig. 5b) as well as the required joint torques show significant discontinuities when the contact situation changes. Adjusting  $\tau_0$  according to (33) allows to make these transitions perfectly smooth.

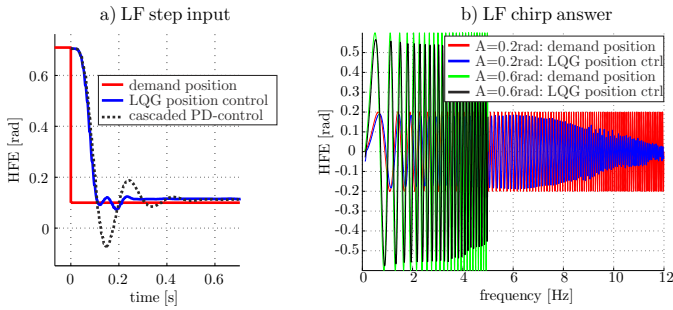


Fig. 6. (a) Step answer: The position control performance with a fast low-level LQG position controller based on a SEA model performs significantly better than the equivalent cascaded controller. (b) Linear chirp time series plotted against frequency: The position control bandwidth drops for large amplitudes due to saturation effects.

## VIII. REAL-TIME IMPLEMENTATION AND HARDWARE EXPERIMENTS

### A. Controller framework and real-time simulation

The presented controller framework is based on analytical kinematics and dynamics. This ensures a very clear separation between controller and simulation environment respectively actual hardware [11]. The simulator part consists of the real-time multi-body-system simulation environment SL [24] that communicates through shared memory with the controller interface. The controller contains the analytical global kinematics and dynamics (Section IV), a realistic state estimator [3], and actuator/sensor models. A change in the communication layer allows switching to the actual robotic system or even running both simultaneously. Four parallel operating CAN bus systems connect to the low level motor controllers and sensor boards. An inertial measurement unit containing accelerometer and gyroscope delivers information about the current state of the main body. These signals are fused with the leg kinematics (contact conditions) to give precise information about the main body position/orientation as well as about ground elevation [3].

In addition to speed, simplicity of the controller implementation, and independence of the simulation environment, using such a setup has several benefits. The probably most important advantage when working with model based systems is that the dynamics behind the simulation are not exactly equal as the analytical model. Differences mainly arise due to the sensor/actuator models (noise, filtering, bandwidth/saturation limitations) and state estimation, different handling of contact interaction or modeling errors/uncertainties (e.g. point feet). Thereby, plant inversion problems that are often hidden when simulation and control model are exactly equal, can be avoided. Using this framework allowed a successful validation and robustness check of the controller properties.

### B. Low-level position control

As motivated through rather theoretical considerations (Fig. 2), a low-level position controller performs better than the corresponding cascaded structure. This fact was investigated in joint step input experiments. As depicted in Fig. 6a, the LQG [10] structure shows the equal rise time (saturation effects) as the cascaded structure but benefits from a much better damping

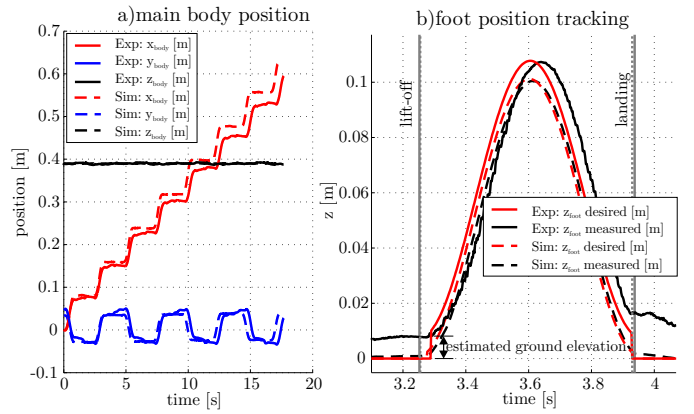


Fig. 7. The simulation and experimental results using exactly the same hierarchical OSC controller implementation with slippage minimization for flat ground agree to a large extent in terms of base trajectory (a) and the quite accurate foot position tracking (b).

behavior and shorter settling time. Analyzing the linear chirp signal (Fig. 6b) shows a bandwidth of close to 10Hz for the linear system dropping down to about 4Hz for very large amplitudes (0.6 rad) due to saturation effects.

### C. Simultaneous walking and contact force optimization

To prove applicability of the proposed methods we finally performed a series of experiments with StarLETH. Static walking on flat terrain including ground contact force optimization showed qualitatively very nice performance using the hybrid structure described in Section V with a combination of low level torque [9] and position [10] controllers. These experiments largely agree with the corresponding simulation of the same controller (Fig. 7a). In addition thereto, the swing leg position tracking based on the low level position controller performs well in both the simulation and the experiment (Fig. 7b). The signals differ since the estimator [3] predicted a minimal ground elevation.

The contact force modulation (Section VI) was tested on a surface with approximately  $40^\circ$  inclination (Fig. 8). Ground truth data of the contact slippage was accessed through vision based foot point tracking in the movies collected during the experiments. As shown in foot position plot of the left-front leg in Fig. 9, activating the contact force alignment significantly lowers slippage. Without an optimal force distribution as a function of the local contact normals, the system fails after a single step sequence (red-dotted), while adjusting the internal forces using the least square method (29) allows to robustly walk on the tree stem (blue-solid).

## IX. CONCLUSION

There are two primary contributions of this paper. First of all, it introduces StarLETH, a compliant quadrupedal robot that was built to study *fast*, *efficient*, and *versatile* locomotion. High compliant series elastic actuation makes the system inherently safe for human and environment interaction, enables the temporary storage of energy, and allows for precise torque control. In return, torque bandwidth limitations arise due to the high compliance and call for appropriate control techniques to ensure fast and precise joint tracking [10]. The introduction

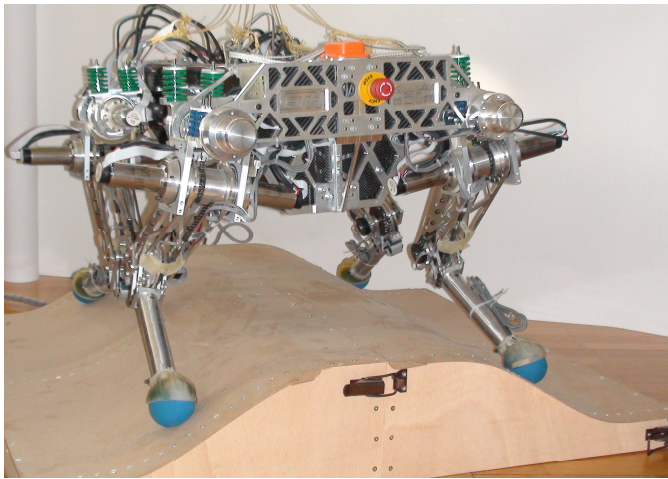


Fig. 8. StarLETH was successfully tested in experiments that required alignment of the contact forces with the actual ground contact normals to avoid slippage.

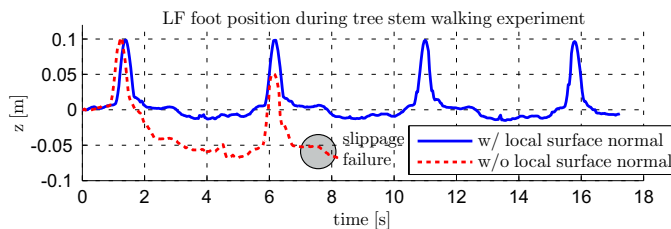


Fig. 9. Walking on the tree stem is only possible through accounting for local surface normal directions (blue-solid). Using a flat ground assumption (red-dotted), the system fails after one full step sequence.

of a hybrid operational space control technique allows to combine the advantages of such local joint position control with global inverse dynamics methods. In the example of quadrupedal walking, it was shown that the application of this method ensures fast tracking of the swing leg (position control) while the rest of the robot is undergoing a hierarchical task decomposition (similar to [27]) that utilizes inverse dynamics to ensure a robust and compliant behavior of the entire system. Additionally, we showed in simulations and hardware experiments, how the contact forces can be aligned to the local surface geometry without altering or impairing the motion. This greatly reduces the risk of slippage, even in highly challenging terrain (Fig. 8).

#### REFERENCES

- [1] R. M. Alexander. 3 uses for springs in legged locomotion. *Int. Journal of Robotics Research*, 9(2):53–61, 1990.
- [2] Inc. Barrett Technology. WAM arm, 2010. URL <http://www.barrett.com/robot/products-arm.htm>.
- [3] M. Bloesch, M. Hutter, M. H. Hoepflinger, C. D. Remy, C. Gehring, and R. Siegwart. State estimation for legged robots - consistent fusion of leg kinematics and imu. *RSS Robotics Science and Systems*, 2012.
- [4] G. A. Cavagna, F. P. Saibene, and R. Margaria. Mechanical work in running. *J Appl Physiol*, 19(2):249–256, 1964.
- [5] P. Gonzalez de Santos, E. Garca, and J. Estremera. *Quadrupedal locomotion: an introduction to the control of four-legged robots*. Springer, 2006.
- [6] G. Hirzinger, A. Albu-Schaffer, M. Hahnle, I. Schaefer, and N. Sporer. On a new generation of torque controlled light-

- weight robots. In *Int. Conf. on Robotics and Automation (ICRA)*, 2001.
- [7] M. H. Hoepflinger, C. David Remy, M. Hutter, L. Spinello, and R. Siegwart. Haptic terrain classification for legged robots. In *Int. Conf. on Robotics and Automation (ICRA)*, 2010.
- [8] M. Hutter, C. Gehring, and R. Siegwart. proNEu: Derivation of analytical kinematics and dynamics. Technical report, Autonomous Systems Lab, ETHZ, 2011.
- [9] M. Hutter, C. David Remy, M. H. Hoepflinger, and R. Siegwart. High compliant series elastic actuation for the robotic leg scarleth. In *Int. Conf. on Climbing and Walking Robots (CLAWAR)*, 2011.
- [10] M. Hutter, C. David Remy, M. H. Hoepflinger, and R. Siegwart. Scarleth: Design and control of a planar running robot. In *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011.
- [11] M. Hutter, C. Gehring, M. Bloesch, M. H. Hoepflinger, C. David Remy, and R. Siegwart. Starleth: a compliant quadrupedal robot for fast, efficient, and versatile locomotion. In *Int. Conf. on Climbing and Walking Robots (CLAWAR)*, 2012.
- [12] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal of Robotics and Automation*, 3(1):43–53, 1987.
- [13] J. Z. Kolter and A. Y. Ng. The stanford little dog: A learning and rapid replanning approach to quadruped locomotion. *The Int. Journal of Robotics Research*, 30(2):150–174, 2011.
- [14] R.B. McGehee. Some finite state aspects of legged locomotion. *Mathematical Biosciences*, 2:67–84, 1968.
- [15] M. Mistry, J. Buchli, and S. Schaal. Inverse dynamics control of floating base systems using orthogonal decomposition. In *Int. Conf. on Robotics and Automation (ICRA)*, 2010.
- [16] N. J. Mitra, A. Nguyen, and L. Guibas. Estimating surface normals in noisy point cloud data. *Int. Journal of Computational Geometry & Applications*, 14(4-5):261–276, 2004.
- [17] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal. Operational space control: A theoretical and empirical comparison. *The Int. Journal of Robotics Research*, 27(6):737–757, 2008.
- [18] R. Philippsen, L. Sentis, and O. Khatib. An open source extensible software package to create whole-body compliant skills in personal mobile manipulators. In *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011.
- [19] G. Pratt and M. Williamson. Series elastic actuators. In *Int. Conf. on Intelligent Robots and Systems (IROS)*, 1995.
- [20] M. Quigley, A. Asbeck, and A. Ng. A low-cost compliant 7-dof robotic manipulator. In *Int. Conf. on Robotics and Automation (ICRA)*, 2011.
- [21] L. Righetti, J. Buchli, M. Mistry, and S. Schaal. Inverse dynamics control of floating-base robots with external constraints: A unified view. In *Int. Conf. on Robotics and Automation (ICRA)*, 2011.
- [22] L. Righetti, J. Buchli, M. Mistry, and S. Schaal. Control of legged robots with optimal distribution of contact forces. In *Int. Conf. on Humanoid Robots (Humanoids)*, 2011.
- [23] Meka Robotics. A2 compliant arm, 2011. URL <http://www.mekabot.com/arm.html>.
- [24] Stefan Schaal. The SL simulation and real-time control software package. Technical report, USC, 2009.
- [25] L. Sentis. *Synthesis and Control of Whole-Body Behaviors in Humanoid Systems*. Ph.d. thesis, 2007.
- [26] L. Sentis. *Compliant Control of Whole-Body Multi-Contact Behaviors in Humanoid Robots*. Springer Global Editorial, 2009.
- [27] L. Sentis and O. Khatib. Control of free-floating humanoid robots through task prioritization. In *Int. Conf. on Robotics and Automation (ICRA)*, 2005.
- [28] D. Zhou, K. H. Low, and T. Zielinska. An efficient foot-force distribution algorithm for quadruped walking robots. *Robotica*, 18(04):403–413, 2000.