

# Using Torque Redundancy to Optimize Contact Forces in Legged Robots

Ludovic Righetti, Jonas Buchli, Michael Mistry,  
Mrinal Kalakrishnan, and Stefan Schaal

**Abstract.** The development of legged robots for complex environments requires controllers that guarantee both high tracking performance and compliance with the environment. More specifically the control of contact interaction with the environment is of crucial importance to ensure stable, robust and safe motions. In the following, we present an inverse dynamics controller that exploits torque redundancy to directly and explicitly minimize any combination of linear and quadratic costs in the contact constraints and in the commands. Such a result is particularly relevant for legged robots as it allows to use torque redundancy to directly optimize contact interactions. For example, given a desired locomotion behavior, it can guarantee the minimization of contact forces to reduce slipping on difficult terrains while ensuring high tracking performance of the desired motion. The proposed controller is very simple and computationally efficient, and most importantly it can greatly improve the performance of legged locomotion on difficult terrains as can be seen in the experimental results.

---

Ludovic Righetti · Stefan Schaal  
University of Southern California, Los Angeles, USA  
Max-Planck Institute for Intelligent Systems, Tübingen, Germany  
e-mail: ludovic.righetti@a3.epfl.ch, sschaal@usc.edu

Jonas Buchli  
Italian Institute of Technology, Genoa, Italy  
e-mail: jonas@buchli.org

Michael Mistry  
University of Birmingham, Birmingham, UK  
e-mail: m.n.mistryl@bham.ac.uk

Mrinal Kalakrishnan  
University of Southern California, Los Angeles, USA  
e-mail: kalakris@usc.edu

## 1 Introduction

We are interested in developing legged robots able to perform difficult tasks in challenging environments. In order to achieve such a goal, we need controllers that can guarantee at the same time good motion tracking performance and an adequate control of the contact interactions with the environment. Tracking performance is important for tasks requiring agility such as climbing or walking on very rough terrain, where the robot feet must be placed at very precise location. A certain degree of compliance is desirable to handle unexpected disturbances. But more importantly, the controller should also be able to directly optimize contact forces. For example, a walking robot should minimize tangential contact forces to avoid slipping while it should control contact forces to redistribute its weight among the different limbs during climbing tasks to increase the range of possible motions. In this chapter, we present an inverse dynamics controller for high tracking performance and compliance that ensures an optimal distribution of contact forces by using the redundancy available in the commands.

Model-based approaches such as inverse dynamics or operational space control offer an interesting framework for the control of legged robots. Indeed, they can greatly improve tracking performance while allowing more compliant control since they require lower error feedback gains. While these methods are standard for manipulators [13], they are not yet widely used for legged robots. Indeed legged robots are different from manipulators fixed to the ground because they are under-actuated due to their floating base<sup>1</sup> and they are subject to changing contact interaction with the environment as their legs move. These differences make the design of inverse dynamics controllers for legged robots more complex.

Recently, several inverse dynamics controllers were proposed for floating-base robots subject to contact constraints. These methods compute required torques without measuring contact forces by assuming idealized contact constraints and projecting the dynamics into a constraint free space [1, 7, 11]. The methods proposed in [7] is of special interest since it does not require a structured representation of the dynamics (i.e. no need to compute individual components like the inertia matrix, Coriolis, and gravity terms) and mainly relies on kinematic quantities, which makes it particularly robust to uncertainties in parameter estimation and, additionally, computationally very effective.

Interestingly when there are more than six contact constraints with the environment, the inverse dynamics problem is under-determined in the sense that there is an infinite number of torque commands for a constraint-consistent desired motion. It is the case, for example, when a biped has its two feet on the ground or when a quadruped robot with point feet has more than 2 feet on the ground. There are more degrees of freedom for the actuation than for the possible motions due to the constraints imposed by the contacts. The over-constrained case is very interesting because there is an infinite number of possible choice of commands to realize a desired motion. In general, redundancy is resolved by minimizing a cost criterion, e.g.,

---

<sup>1</sup> We refer to floating base robots that are not fixed to the ground due to their 6 non-actuated DOFs that describe the position and orientation of the robot relative to an inertial frame.

a quadratic cost in the commands as in [7, 10]. We will see in the following that we can use torque redundancy to directly manipulate the contact forces instead. It will allow us to create tracking controllers optimal with respect to any combination of linear and quadratic costs of the contact forces and the commands.

The problem of contact force distribution during locomotion has already been addressed in the literature [4,6]. Force control approaches have been proposed [3,14] to directly control the contact forces of a legged robot. These approaches have been successfully used for the control of biped robots. In general, the manipulation of contact forces is done as a primary goal, for example to create balance controllers. A potential desired motion is then treated as a secondary goal. It is in contrast with our approach that aims at creating tracking controllers able to manipulate contact forces using torque redundancy, i.e. trajectory tracking is the primary goal of the controller.

The manipulation of contact forces in the context of inverse dynamics or operational space control has also been addressed in [12]. In this case, desired contact forces are explicitly controlled using the torques acting in the nullspace of the motion. Such an approach is interesting if one has a precise objective for the contact forces. However it does not provide any optimality result on the force distribution.

Recently we have been investigating how we could use torque redundancy in inverse dynamics controllers and operational space controllers to optimize contact forces [8,9]. In this chapter, we present some of the latest developments of our work. We show how we can create inverse dynamics controllers that are optimal with respect to any combination of linear and quadratic cost in the constraint forces and in the commands. The controller is computationally simple and robust to parameter estimation errors, which make it well suited for high performance control of complex robots with a large number of degrees of freedom, such as humanoid robots. We present several applications of the controller to legged robots. Experimental results show that for the same desired motion, the use of torque redundancy can significantly improve the performance of the robot during locomotion. It is worth mentioning that several additional applications and technical details of this work can be found in [8].

## 2 Problem Formulation

We first consider the general problem of inverse dynamics for floating-base robots under constraints. First we present the constraint model and its assumptions. Then we present the control law recently derived by Mistry et al. [7] for legged robots and that we will use in subsequent sections.

### 2.1 *Rigid Body Dynamics Model*

Assuming that the robot in contact with its environment obeys rigid body dynamics, its equations of motion are

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{h} = \mathbf{S}^T \boldsymbol{\tau} + \mathbf{J}_c^T \boldsymbol{\lambda} \quad (1)$$

under the  $k$  constraints

$$\mathbf{J}_c \ddot{\mathbf{q}} = \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) \quad (2)$$

where  $\mathbf{q} = [\mathbf{x}_j^T \ \mathbf{x}_b^T]^T$  is the vector of joint positions ( $\mathbf{x}_j \in \mathbb{R}^n$ ) and base positions and orientations ( $\mathbf{x}_b \in \text{SE}(3)$ ),  $\mathbf{M} \in \mathbb{R}^{(n+6) \times (n+6)}$  is the rigid body dynamics inertia matrix,  $\mathbf{h} \in \mathbb{R}^{n+6}$  is a generalized force vector containing all the modeled forces, including the Coriolis, centrifugal and gravitational forces as well as friction forces in the joints.  $\boldsymbol{\tau} \in \mathbb{R}^n$  is the actuation vector and  $\mathbf{S} = [\mathbf{I}_{n \times n} \ 0_{n \times 6}] \in \mathbb{R}^{n \times (n+6)}$  is the joint selection matrix that characterize the under-actuation.  $\mathbf{J}_c \in \mathbb{R}^{k \times (n+6)}$  is the Jacobian of the  $k$  constraints with the  $\boldsymbol{\lambda} \in \mathbb{R}^k$  Lagrange multipliers that correspond to the constraint forces.

Following the ideas from [15] we expressed the constraints in acceleration form (i.e. as given in Eq. 2). Holonomic constraints can be expressed by differentiating them twice and non-holonomic constraints by differentiating them once. We assume in the following, without loss of generality, that  $\mathbf{J}_c$  is full row rank, in the sense that all constraints are linearly independent. If it is not the case then one can easily find a reduced number of independent constraints, for example by using the SVD decomposition of  $\mathbf{J}_c$ .

*Example 1.* If we assume that the position of the point feet of a legged robot are given by  $\mathbf{x}_c$ , then the constraints that the feet do not move relative to the ground can be written as  $\mathbf{x}_c = \text{constant}$  or equivalently by  $\dot{\mathbf{x}}_c = 0$ . Relating this to the motion of the joints of the robot using the Jacobian of  $\mathbf{x}_c$  we have  $\mathbf{J}_c \dot{\mathbf{q}} = \dot{\mathbf{x}}_c = 0$ , which we differentiate once again to get  $\mathbf{J}_c \ddot{\mathbf{q}} = -\dot{\mathbf{J}}_c \dot{\mathbf{q}}$ .

We assume that the movement plan of the robot is expressed by desired joint accelerations that are constraint consistent, i.e., Equation (2) with  $\ddot{\mathbf{q}} = \ddot{\mathbf{q}}_d$  holds. These accelerations will be satisfied if and only if they are of the form

$$\ddot{\mathbf{q}}_d = \mathbf{J}_c^G \mathbf{b} + (\mathbf{I} - \mathbf{J}_c^G \mathbf{J}_c) \ddot{\mathbf{q}}_0 \quad (3)$$

where  $\mathbf{J}_c^G$  can be any generalized inverse [2] of  $\mathbf{J}_c$ , i.e. a matrix such that  $\mathbf{J}_c \mathbf{J}_c^G \mathbf{J}_c = \mathbf{J}_c$ . Here,  $\ddot{\mathbf{q}}_0$  is an arbitrary acceleration vector and  $(\mathbf{I} - \mathbf{J}_c^G \mathbf{J}_c)$  projects these accelerations into the null space of the constraints.

The general problem of inverse dynamics is then to compute the torques  $\boldsymbol{\tau}$  such that they will achieve the desired accelerations  $\ddot{\mathbf{q}}_d$ . The range of unconstrained movements lies in a  $n + 6 - k$  dimensional space while the dimension of the control vector is  $n$ . Therefore we can distinguish three cases for inverse dynamics depending on the number of constraints:

- $k < 6$ , the system is underactuated since there are more dimensions of movement than dimensions of actuation. There is at most one solution to the inverse dynamics problem: for a solution to exist, the desired accelerations must not only be constraint consistent, they moreover need to be consistent with the dynamics of Eq. (1). For example one can think of the case of no constraints, when a cat is falling and cannot orient its body independently from moving its joints.

- $k = 6$ , the system is fully actuated. There is exactly one solution provided that the desired accelerations are constraint consistent. This case is similar to the inverse dynamics problem of a manipulator fixed to the ground.
- $k > 6$ , the system is overconstrained. There is an infinite number of solutions for  $\tau$  that will achieve perfect tracking of  $\ddot{\mathbf{q}}_d$ . It is the case, for example, when a humanoid has both feet flat on the ground, or with one foot and one hand in flat contact with the environment or when a quadruped with point feet has more than two feet on the ground.

In the following, we only consider the overconstrained case ( $k > 6$ ) since it is the only case where torque redundancy can be used to optimize constraint forces.

## 2.2 Inverse Dynamics Solution Using Orthogonal Projections

By using orthogonal projections, Mistry et al. [7] proposed recently an efficient way to compute the inverse dynamics of a constrained under-actuated system without the need to measure contact forces. More precisely the authors use the QR decomposition of the constraint Jacobian  $\mathbf{J}_c^T = \mathbf{Q} [\mathbf{R}^T \mathbf{0}]^T$ , where  $\mathbf{Q} \in \mathbb{R}^{(n+6) \times (n+6)}$  is an orthogonal matrix (i.e.  $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}$ ) and  $\mathbf{R} \in \mathbb{R}^{6 \times 6}$  is an upper triangular invertible matrix. If we decompose  $\mathbf{Q} = [\mathbf{Q}_c \mathbf{Q}_u]$  into the constrained,  $\mathbf{Q}_c \in \mathbb{R}^{(n+6) \times k}$ , and unconstrained,  $\mathbf{Q}_u \in \mathbb{R}^{(n+6) \times (n+6-k)}$ , components, the general solution for the inverse dynamics torques given desired accelerations  $\ddot{\mathbf{q}}_d$  can be written as

$$\tau(\mathbf{W}, \tau_0) = \overline{\mathbf{Q}_u^T \mathbf{S}^T} \mathbf{Q}_u^T (\mathbf{M} \ddot{\mathbf{q}}_d + \mathbf{h}) + (\mathbf{I} - \overline{\mathbf{Q}_u^T \mathbf{S}^T} \mathbf{Q}_u^T \mathbf{S}^T) \mathbf{W}^{-1} \tau_0 \quad (4)$$

with generalized inverse

$$\overline{\mathbf{Q}_u^T \mathbf{S}^T} = \mathbf{W}^{-1} \mathbf{S} \mathbf{Q}_u (\mathbf{Q}_u^T \mathbf{S}^T \mathbf{W}^{-1} \mathbf{S} \mathbf{Q}_u)^{-1} \quad (5)$$

where  $\mathbf{W} \in \mathbb{R}^{n \times n}$  is a symmetric positive definite matrix and  $\tau_0$  is an arbitrary internal torque –  $\tau_0$  is premultiplied by a projection matrix that guarantees that  $\tau_0$  can only create internal forces, but no movement.

Moreover the resulting constraint forces can be predicted by

$$\lambda = \mathbf{R}^{-1} \mathbf{Q}_c^T (\mathbf{M} \ddot{\mathbf{q}}_d + \mathbf{h} - \mathbf{S}^T \tau) \quad (6)$$

*Remark 1.* In the case where  $k = 6$ , there is only one solution and  $\overline{\mathbf{Q}_u^T \mathbf{S}^T} = (\mathbf{Q}_u^T \mathbf{S}^T)^{-1}$  and the nullspace is empty. When  $k < 6$ , there is at most one solution and  $\overline{\mathbf{Q}_u^T \mathbf{S}^T} = (\mathbf{Q}_u^T \mathbf{S}^T)^+$ , where  $()^+$  denotes the Moore-Penrose generalized inverse and the nullspace is also empty.

We note that the torque control law  $\tau(\mathbf{W}, \tau_0)$  is parametrized by a weight matrix and an internal torque vector. It is this parameterization that can be exploited to resolve torque redundancy. One result that is already known is that  $\tau_0 = \mathbf{0}$  leads to the minimization of the cost  $\tau^T \mathbf{W} \tau$  at each instant of time [7]. We show in the following how we can do the same for costs in the *constraint forces*.

### 3 Minimization of Constraint Forces

In the previous section, we have presented an idealized way of describing constraints using acceleration equalities. However using equalities offers a limited representational capability and cannot capture important aspects such as physical limitations on the constraint forces that can be generated. For example, in the context of a constraint that enforces a foot to stay on the ground, we cannot represent the fact that ground reaction forces should be inside the cone of friction to avoid slipping, which means that the ratio between forces tangential and normal to the contact must satisfy an inequality. We can immediately see that including such inequalities for the inverse dynamics problem will have the consequence that we will not be able to solve the inverse dynamics problem without a (possibly complex) iterative optimization algorithm. This is in contrast to the simple analytical solution presented in Eq. (4).

Another solution would be to directly minimize a cost in the constraint forces that would take into account those inequalities implicitly. If we can get the torque distribution that minimizes such a cost, then, implicitly, it will try to find a solution that tends to fulfill the inequalities. Another advantage is that one does not need to know the exact model of the contact, e.g., the friction cone. For example minimizing a cost that penalizes the tangential forces during contact will ensure that the robot minimizes slipping for all sizes of friction cones, i.e., the controller will act as conservative as possible towards slipping. While it is possible that such an approach finds a solution that violates the constraints even if a correct solution exists, we will see in the next section that it is rarely a problem in practice. A detailed discussion on limitations and advantages of this approach can be found in [8].

#### 3.1 General Result

We now present the general result used to construct controllers that minimize any combinations of linear and quadratic costs in the contacts and in the commands.

**Theorem 1.** *The inverse dynamics controller (Eq.4) that minimizes the cost*

$$\frac{1}{2}\boldsymbol{\tau}^T \mathbf{W}_\tau \boldsymbol{\tau} + \mathbf{b}_\tau^T \boldsymbol{\tau} + \frac{1}{2}\boldsymbol{\lambda}^T \mathbf{W}_\lambda \boldsymbol{\lambda} + \mathbf{b}_\lambda^T \boldsymbol{\lambda} \quad (7)$$

is chosen by setting

$$\mathbf{W} = \mathbf{W}_\tau + \mathbf{S}\mathbf{W}_c\mathbf{S}^T \quad (8)$$

$$\boldsymbol{\tau}_0 = -\mathbf{b}_\tau + \mathbf{S}\mathbf{W}_c(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{h}) + \mathbf{S}\mathbf{b}_c \quad (9)$$

where

$$\mathbf{W}_c = \mathbf{Q} \begin{bmatrix} \mathbf{R}^{-T} \mathbf{W}_\lambda \mathbf{R}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{Q}^T, \quad (10)$$

$$\mathbf{b}_c = \mathbf{Q} \begin{bmatrix} \mathbf{R}^{-T} \\ \mathbf{0} \end{bmatrix} \mathbf{b}_\lambda, \quad (11)$$

$\mathbf{J}_c^T = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ 0 \end{bmatrix}$  is the QR decomposition of the constraint Jacobian and  $\mathbf{W}_\lambda$  is such that  $\mathbf{W}_\tau + \mathbf{S}\mathbf{W}_c\mathbf{S}^T$  is symmetric positive definite.

We omit the proof as it can be found in [8] with more technical details. The result provides us with a way to directly and explicitly optimize at the same time both command costs and constraint forces using torque redundancy. The result is very general as it can be applied for any type of constraints expressed in the form of Equation (2) and for any combination of linear and quadratic costs of those constraints. The result applies to any control system whose equations of motion can be written as Equation (1) and is therefore not limited to legged robots.

*Remark 2.* In addition to its very nice computational properties, the formulation of the inverse dynamics control as proposed in [7] is very convenient as we can use the same orthogonal decomposition  $\mathbf{Q}$  to derive the torque parametrization for optimal distribution of constraint forces.

Now cost functions can be designed depending on the desired application in order to manipulate the generated constraint forces and torques.

## 3.2 Examples of Optimization

In this section we discuss a few controllers that can be derived using the results of the previous section and some of their properties relevant for applications.

### 3.2.1 Minimization of Tangential Contact Forces

To ensure proper contact with the ground, one has to guarantee that the ground reaction forces stay within the friction cones. The friction cone is a purely geometric constraint that is defined by a friction constant and the orientation of the contact surface with the contacting foot. In the case of locomotion, to avoid slipping, one would like to have the reaction forces as orthogonal to the constraint surface as possible. In other words, the tangential forces should be minimized. Moreover in the case of a flat foot on the ground, the resulting moment around the foot should be as small as possible. The cost to optimize should therefore take into account the orientation of the ground in order to redirect contact forces in a more desirable direction.

In order to minimize the tangential forces and the moments around the foot, we propose the following cost

$$\mathbf{W}_\lambda = \begin{bmatrix} \mathbf{R}_{leg_1}^T \mathbf{W}_{leg_1} \mathbf{R}_{leg_1} & & 0 \\ & \ddots & \\ 0 & & \mathbf{R}_{leg_n}^T \mathbf{W}_{leg_n} \mathbf{R}_{leg_n} \end{bmatrix} \quad (12)$$

with

$$\mathbf{W}_{leg_i} = \text{diag}(K_{tx}, K_{ty}, 1, K_{mx}, K_{my}, K_{mz}) \quad (13)$$

where  $\mathbf{R}_{legn}$  is a rotation matrix that corresponds to the orientation of the surface with respect to the inertial frame (i.e. it aligns the reaction forces and moments of a foot with the orientation of the surface).  $K_t$  are gains associated with the tangential forces in x,y directions and the moments in x,y,z directions. Here we assume that the z direction is normal to the ground.

*Remark 3.* In the case of point feet the matrix  $\mathbf{W}_{leg}$  is reduced to its  $3 \times 3$  upper-left sub-matrix. The gains  $K_t$  and  $K_m$  must be chosen with appropriate units such that the summations in the cost are unit consistent. The exact values of the gains need to be designed specific to the application.

### 3.2.2 Tracking Desired Contact Forces

Instead of only minimizing constraint forces, we can also manipulate the value of these forces explicitly in order to create more interesting contacts. For example, when one wants to manipulate the center of pressure of the feet during walking (in a case where the feet are not co-planar) or to explicitly regulate force interaction in specified directions. The previous results allow us to do that very easily. Assume we have a desired interaction force vector  $\lambda_d$  and we want the contact forces to follow this desired force vector as close as possible. We can then minimize a weighted square error that measures the performance in tracking

$$\frac{1}{2}(\lambda - \lambda_d)^T \mathbf{K}(\lambda - \lambda_d) \quad (14)$$

where  $\mathbf{K}$  is symmetric positive definite. Then, noticing that

$$\frac{1}{2}(\lambda - \lambda_d)^T \mathbf{K}(\lambda - \lambda_d) = \frac{1}{2}\lambda^T \mathbf{K}\lambda + \frac{1}{2}\lambda_d^T \mathbf{K}\lambda_d - \lambda_d^T \mathbf{K}\lambda \quad (15)$$

we can create a controller that will optimally track those constraint forces in the sense that it will minimize the tracking error by choosing the following controller parameters

$$\mathbf{W}_\lambda = \mathbf{K} \quad (16)$$

$$\mathbf{b}_\lambda = -\lambda_d^T \mathbf{K} \quad (17)$$

It is interesting since we can use the torque redundancy to explicitly manipulate contact forces and track desired forces. The tracking performance will obviously be dependent on the desired forces and the redundancy left in the nullspace of the motion. It means that it will not be possible to track arbitrary forces, however we are guaranteed to have an optimal performance in terms of the tracking cost.

## 4 Experimental Results

We now present some experiments in order to illustrate the advantages of using the previously proposed redundancy resolution scheme. In the following, we apply our



**Fig. 1** The robots used in the simulations: left, SARCOS humanoid, right, Boston Dynamics Little Dog

approach on two different simulated robots (Figure 1) using different optimization criteria for the controller as well as a preliminary application on a real quadruped robot.

#### ***4.1 Control of a Humanoid Robot***

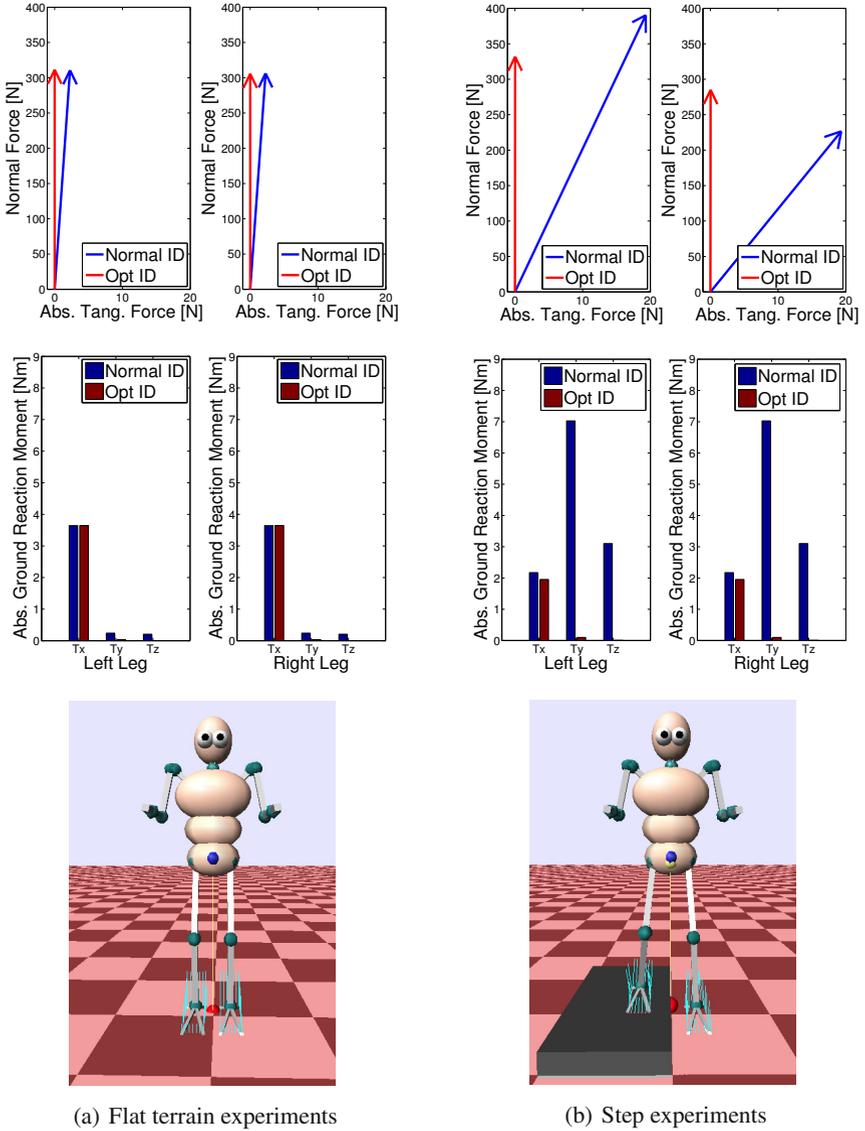
Our first results are shown on a simulation of the Sarcos humanoid robot (Figure 1) which is a 34-DOFs human size torque controlled humanoid robot. The main objective of the experiment is to illustrate in simple examples how the inverse dynamics controller can manipulate contact forces.

In the experiment, we keep the robot in a desired posture while minimizing tangential ground reaction forces together with the moment generated around the feet (as described in Section 3.2.1). We compare the resulting contact forces with the original inverse dynamics controller that minimizes the total command cost  $\tau^T \tau$ . In this static case, the contact forces are only generated through actuation and gravity.

We tested both controllers in two contexts: in a symmetric posture on a flat terrain and on a terrain where the robot is stepping on a 10 cm box. The results of the experiments are shown in Figure 2. We notice that on the flat surface, both controllers generate similar contact forces and moments. However in the asymmetric case, when the robot is on a step, we clearly see a difference in force and moment distributions. The original controller generates unnecessary tangential forces and moments to keep its posture while the other controller minimizes the contact constraints. We can clearly see the advantage of such a controller when stepping for example on an object that is not bolted to the ground and could potentially move.

#### ***4.2 Simulation of Quadruped Locomotion***

Next, we show the performance of our torque controller in a more dynamic situation where we used a simulation of the LittleDog robot for quadruped locomotion. The



**Fig. 2** Results of the first experiment with the humanoid simulation. We depict the contact forces split into normal forces and absolute tangential forces (upper graphs) and show the 3 moments created around the feet (middle graph) and a snapshot of the posture of the robot in each experiment. The regular inverse dynamics controller is plotted in blue and the controller minimizing contact forces is plotted in red. For each experiment we set  $K_{rx} = K_{ry} = 1000N^{-2}$  and  $K_{mx} = K_{my} = K_{mz} = 2000Nm^{-2}$ .

planning of desired joint positions, velocities and accelerations is done following the method proposed in [5]. We use a ZMP-based algorithm to plan the desired center of gravity (COG) motion of the robot. The COG plan is completed with a world space target for the position of the foot of the swing leg. These kinematic trajectories are converted into joint space reference trajectories via an analytical inverse kinematics model, which exists for this robot due to its 3 DOF legs.

In order to achieve asymptotically stable tracking of trajectories in joint space, an error feedback command in joint-space is added to the feedforward command computed by the inverse dynamics law. The resulting reference command, given desired  $\mathbf{q}_d$ ,  $\dot{\mathbf{q}}_d$  and  $\ddot{\mathbf{q}}_d$ , is therefore

$$\boldsymbol{\tau} = \boldsymbol{\tau}(\mathbf{W}, \boldsymbol{\tau}_0) + \mathbf{PID}(\mathbf{q}_d, \dot{\mathbf{q}}_d) \quad (18)$$

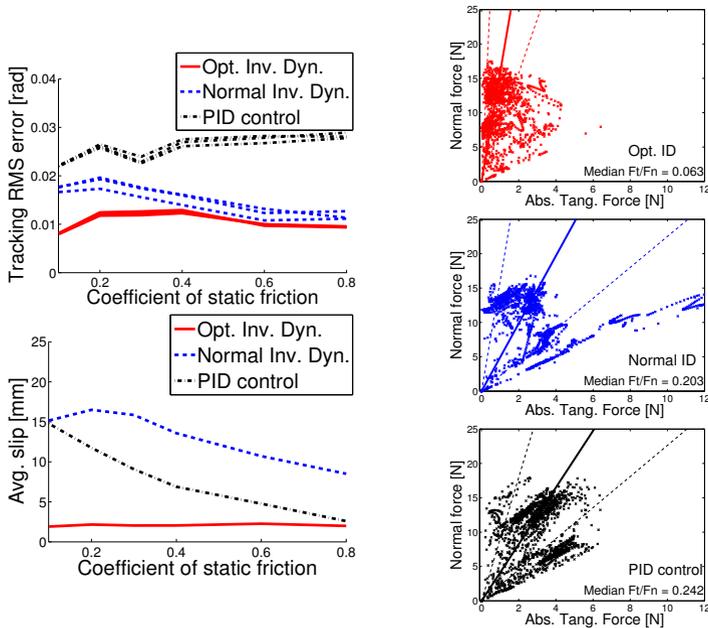
where **PID** corresponds to a joint space PID error feedback controller.

We use a physical simulation of the Little Dog robot (Fig. 1). In order to stay as close as possible to reality, we simulate the controller as it would be executed on the real robot. The real robot has an on-board controller running at 400 Hz that generates the PID commands and can add a feedforward torque command (i.e. the inverse dynamics torque in our case). The desired positions, velocities and feedforward commands are generated on a host computer in a different controller that is running at 100 Hz. It must be noted that the inverse dynamics controller is therefore running at a relatively slow bandwidth for torque control and is much slower than the PID control loop. Therefore it can have a negative effect on the actual performance of the simulated controllers as opposed to what would be produced by an idealized or perfect model.

In order to show the performance of our method, we tested the locomotion of the robot with the controller we proposed as compared to two other controllers. First the original inverse dynamics controller that minimizes the total torque command  $\boldsymbol{\tau}^T \boldsymbol{\tau}$ , and second a PID controller without inverse dynamics but with gains twice as high as the other controllers. The higher gains are required to maintain sufficient nominal tracking performance.

We systematically tested the performance of locomotion for these three controllers on a flat, level surface with different coefficients of static friction and on a 0.25 radians sloped surface. For each of the experiments, we measured the tracking performance by computing the root mean square (RMS) tracking error in joint space. Furthermore, the distribution of ground reaction forces at each leg is recorded and the average amount of leg slipping per stance phase is computed.

In Figures 3 and 4, we illustrate the results of these experiments. We show the results only for the front left leg since the results for the other legs are qualitatively the same. We can notice that in all the experiments, the controller we proposed always achieves the best performance in terms of both low tracking error and low slipping. As expected, we see better tracking performance for both inverse dynamics controllers with low PID gains as compared to the high-gain PID controller. We must also note that, in addition, we realized a *compliant control* of the robots, which is not possible for the PID controller.

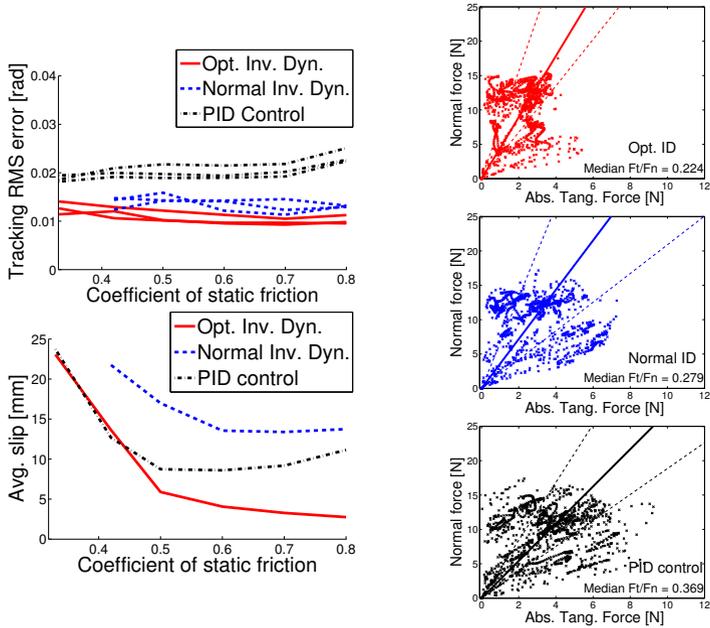


**Fig. 3** Experiments on flat terrain. The figures on the left show the performance measures for the 3 different controllers (cf. text). The figures on the right show the distribution of the contact forces normal to the terrain as a function of the absolute value of the corresponding tangential contact forces. The solid lines indicates the median of the distribution and the dashed line the interquartile range of the distribution (i.e. 50% of the distribution lies between the dashed lines). For the controller minimizing contact forces we used  $K_{ix} = K_{iy} = 10N^{-2}$  in all experiments.

On flat ground, our controller minimizing tangential forces leads to very little slip (approx. 2mm), which for practical purposes can be viewed as negligible. There are two sources that lead to this tracking error despite a perfect rigid body model used in the controller and a deterministic simulation. First, the assumption of holonomic ideal constraints is not fulfilled since the simulation uses a penalty method (i.e. a spring-damper model) to model ground contact and friction. Second, there might be some numerical inaccuracies building up due to numerical integration, even though this can generally be assumed to be very small.

It should be noted that the amount of slip for the other controllers increases as the friction coefficient is lowered and goes up to more than 1.5 cm for the lowest chosen friction coefficient of 0.1. It seems that the PID controller degrades less than the normal inverse dynamics when lowering friction.

We also see that the typical distribution of ground reaction forces is more vertical for the controller optimizing the tangential forces and that the variation of this distribution is also lower. We can therefore conclude that the controller generates contact forces that are oriented more suitably for walking without slipping.



**Fig. 4** Experiments on the slope. The figures on the left show the performance measures for the 3 different controllers (cf. text). The figures on the right show the distribution of the contact forces normal to the terrain as a function of the absolute value of the corresponding tangential contact forces. The solid lines indicates the median of the distribution and the dashed line the interquartile range of the distribution (i.e. 50 % of the distribution lies between the dashed lines). For the controller minimizing contact forces we used  $K_{rx} = K_{ry} = 10N^{-2}$  in all experiments.

On the slope, the results of the performance measured by the slip are even more distinct. The controller having optimal distribution of forces slips less than 6 mm for a friction coefficient higher than 0.5, while the other controllers slip up to 2 to 3 times more for the same simulation conditions. We note that the robot controlled using the original inverse dynamics controller was not able to climb slopes with static friction lower than 0.42 (i.e. the robot would slip and eventually fall) while the other controllers could climb a slope with static friction as low as 0.33. The amount of slipping is still lower for the proposed controller but we note that the high gain PID controller has similar performance for the lowest friction coefficient – again we would like to point out that, however, the PID controller is not compliant and rather stiff.

Again, the distribution of forces on the ground varies less and is more vertically oriented in the case of the controller using the optimal distribution of forces.

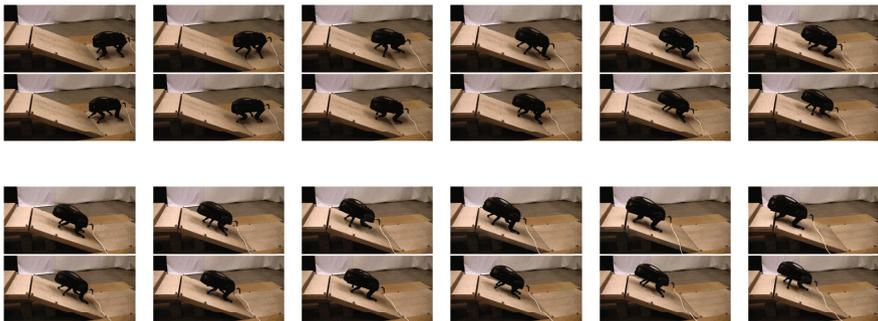
### 4.3 Application to Real Robot

In this section we present experiments with the real Little Dog robot. We ran the locomotion controller with the original inverse dynamics controller minimizing the torque command and with the new controller optimizing tangential ground reaction forces. The terrain was composed of a level flat board and a slope of 0.46 radians, which is higher than the experiments done in simulation – the actual robot turned out to be more capable than our physical simulator. The controller was run at 3 different speeds with a stepping period between 3 to 4 seconds. We were not able to see significant behavioral differences, i.e. we could not see a case where one controller was able to make the robot go up the slope while the other would not. However, we consistently noticed that when using the controller minimizing the tangential forces the robot would reach the top of the slope faster as can be seen in Figure 5. We observed this behavior in all the experiments we ran. It is interesting since the planned desired trajectories were the same for both controllers.

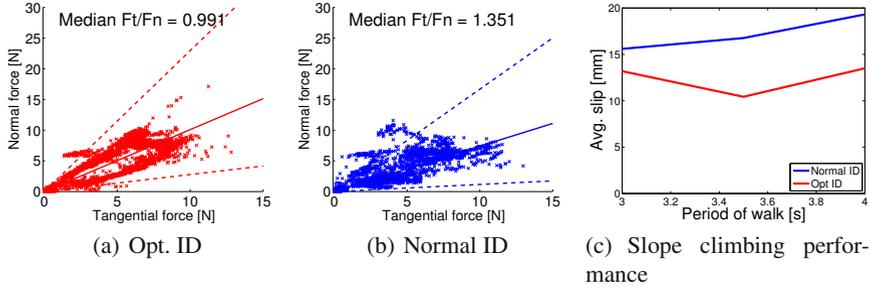
When looking at the amount of slipping, we can see a consistent decrease in the amount of slipping with the controller using optimal distribution of contact forces, as we show in Figure 6. The amount of slip is computed accurately thanks to a motion capture system that tracks the position of the robot. We notice that the robot is slipping on average 30% less when using the new controller.

We also looked at the distribution of contact forces when the robot was walking on level ground (Figure 6). While this data has to be interpreted cautiously in view of the high level of noise in the force measurements, we clearly see a trend for a better distribution of forces when using the new proposed controller.

While we see a clear trend of improvement of locomotion on the real robot, this improvement was not sufficient enough to be able to see the behavioral differences



**Fig. 5** Snapshots of a typical experiment. The figure is organized as two rows of snapshot, time going from left to right. There is approximately 2.5 seconds between each frame. For each frame, the upper graph is the experiment using the controller with minimization of tangential forces and the lower one corresponds to the controller minimizing the command cost. We notice that due to a reduced amount of slipping, the robot using the minimization of tangential forces reaches the top of the slope faster. The planned desired trajectories are the same for both robots.



**Fig. 6** Real robot on 0.25 radians slope, in red the optimal controller in blue the normal ID controller

observed in simulation. However a few limitations of the robotic platform can explain the quality of the results:

- As we discussed in the previous section, our inverse dynamics controller is running at a 100Hz bandwidth on a host computer. Such a low bandwidth of control clearly limits the possible performance of the robot. For example, on a modern torque controlled humanoid robot such as the Sarcos humanoid, one can expect a 1kHz bandwidth of control.
- There are no torque sensors on the robot to close a torque feedback loop – torque control is inferred from current control. Therefore any error in the model converting torques into motor currents will have a negative impact on the actual torques applied to the robot compared to the desired ones.
- The quality of the dynamics model can also play a role. We evaluated the dynamics model of the robot in a similar way we did in the previous section, however it turns out that the dynamics is mainly dominated by friction in the joints, which is a local (decentralized) effect that cannot be re-distributed in a way as suggested by our controller that exploits actuation redundancy. We note that in our experiment roughly 40 to 50% of the total command is due to the PD controller, which is rather high.

These limitations are not fundamental ones and can easily be overcome in more advanced robotic platforms with joint-level force sensing and high control bandwidth. Our current results then suggest that the proposed controller should perform even better on these platforms.

## 5 Conclusion

In this chapter we showed how torque redundancy could be exploited to optimize contact forces in inverse dynamics controllers. The proposed redundancy resolution scheme allows to minimize any combination of linear and quadratic costs in the contact constraints and the commands. Given a desired trajectory, the controller is optimal at each instant of time. The resulting controller is surprisingly simple as it

merely involves the inclusion of a weighted pseudo-inverse and an internal torque vector in the nullspace of the motion. It can therefore be implemented even on real-time computing hardware with modest computational power. Moreover it can be shown that the same controller can be used for whole-body controllers based on the operational space control framework [8].

We proposed to use this result either to track desired contact force trajectories or to minimize tangential contact forces during legged locomotion. Simulation results show that, given desired trajectories, we can exploit torque redundancy to achieve high tracking performance while guaranteeing a better distribution of contact forces and therefore better locomotion on difficult terrains. This constitutes an interesting complement to planning algorithms from the control point of view. Our results on the Little Dog robot, which is not an ideal platform for torque control, also show that the proposed controller is not a pure theoretical result but is realistic enough to be used on real systems. Such results are very encouraging as we expect to see much more improvement on a properly torque-controlled platform.

**Acknowledgements.** This research was supported in part by National Science Foundation grants ECS-0326095, IIS-0535282, IIS-1017134, CNS-0619937, IIS-0917318, CBET-0922784, EECS-0926052, CNS-0960061, the DARPA program on Advanced Robotic Manipulation, the Army Research Office, the Okawa Foundation, the ATR Computational Neuroscience Laboratories, and the Max-Planck-Society.

## References

1. Aghili, F.: A unified approach for inverse and direct dynamics of constrained multi-body systems based on linear projection operator: Applications to control and simulation. *IEEE Transactions on Robotics* 21(5), 834–849 (2005)
2. Ben-Israel, A., Greville, T.: *Generalized inverses: theory and applications*. Springer-Verlag New-York Inc. (2003)
3. Hyon, S., Hale, J.G., Cheng, G.: Full-body compliant human-humanoid interaction: Balancing in the presence of unknown external forces. *IEEE Trans. on Robotics* 23(5), 884–898 (2007)
4. Jiang, W.Y., Liu, A.M., Howard, D.: Optimization of legged robot locomotion by control of foot-force distribution. *Transactions of the Institute of Measurement and Control* 26(4), 311–323 (2004)
5. Kalakrishnan, M., Buchli, J., Pastor, P., Mistry, M., Schaal, S.: Learning, planning, and control for quadruped locomotion over challenging terrain. *International Journal of Robotics Research* 30, 236–258 (2011)
6. Klein, C., Kittivatcharapong, S.: Optimal force distribution for the legs of a walking machine with friction cone constraints. *IEEE Transactions on Robotics and Automation* 6(1), 73–85 (1990)
7. Mistry, M., Buchli, J., Schaal, S.: Inverse dynamics control of floating base systems using orthogonal decomposition. In: *Proc. of the 2010 International Conference on Robotics and Automation* (2010)
8. Righetti, L., Buchli, J., Mistry, M., Kalakrishnan, M., Schaal, S.: Optimal distribution of contact forces with inverse dynamics control (submitted)

9. Righetti, L., Buchli, J., Mistry, M., Schaal, S.: Control of legged robots with optimal distribution of contact forces. In: 2011 11th IEEE-RAS International Conference on Humanoid Robots (Humanoids), pp. 318–324 (2011)
10. Righetti, L., Buchli, J., Mistry, M., Schaal, S.: Inverse Dynamics Control of Floating-Base Robots with External Constraints: a Unified View. In: Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA 2011), pp. 1085–1090 (2011)
11. Sentis, L.: Synthesis and control of whole-body behaviors in humanoid systems. Ph.D. thesis, Stanford University (2007)
12. Sentis, L., Park, J., Khatib, O.: Compliant control of multi-contact and center of mass behaviors in humanoid robots. *IEEE Transactions on Robotics* 26(3), 483–501 (2010)
13. Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G.: *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing. Springer, London (2009), doi:10.1007/978-1-84628-642-1
14. Stephens, B., Atkeson, C.: Dynamic balance force control for compliant humanoid robots. In: International Conference on Intelligent Robots and Systems, IROS (2010)
15. Udwadia, F., Kalaba, R.: On the foundations of analytical dynamics. *Int. J. of Non-Linear Mechanics* 37, 1079–1090 (2002)