Control of legged robots with optimal distribution of contact forces

Ludovic Righetti, Jonas Buchli, Michael Mistry and Stefan Schaal

Abstract—The development of agile and safe humanoid robots require controllers that guarantee both high tracking performance and compliance with the environment. More specifically, the control of contact interaction is of crucial importance for robots that will actively interact with their environment. Model-based controllers such as inverse dynamics or operational space control are very appealing as they offer both high tracking performance and compliance. However, while widely used for fully actuated systems such as manipulators, they are not yet standard controllers for legged robots such as humanoids. Indeed such robots are fundamentally different from manipulators as they are underactuated due to their floating-base and subject to switching contact constraints. In this paper we present an inverse dynamics controller for legged robots that use torque redundancy to create an optimal distribution of contact constraints. The resulting controller is able to minimize, given a desired motion, any quadratic cost of the contact constraints at each instant of time. In particular we show how this can be used to minimize tangential forces during locomotion, therefore significantly improving the locomotion of legged robots on difficult terrains. In addition to the theoretical result, we present simulations of a humanoid and a quadruped robot, as well as experiments on a real quadruped robot that demonstrate the advantages of the controller.

I. INTRODUCTION

In order to build humanoid robots able to interact with their environment, we need controllers that guarantee both high tracking performance and compliance. Tracking performance is required for tasks requiring precision and agility, such as locomotion on difficult terrains or fine manipulation of objects. Compliance is desirable for safe interactions with humans and during contact interaction with the environment to absorb unexpected impacts. In addition, the control of contact interactions is of crucial importance in order to optimize interaction forces. For example, the minimization of tangential contact forces during locomotion will reduce the chance of slipping and therefore improve the performance of the robot on difficult terrains.

Traditional high gain error feedback controllers are problematic to achieve those goals as high tracking performance is directly related to high feedback gains and therefore little compliance. On the other hand, model-based approaches such as inverse dynamics control have proved to be very useful to achieve both high tracking performance and compliance on fully actuated systems such as manipulators. However, legged robots are inherently different from manipulators as they are underactuated due to their floating base\(^1\) and subject to switching contact constraints.

Recently, several approaches for model-based control of floating-base robots subject to contact constraints have been proposed, whether for inverse dynamics [1], [2] or whole-body operational space control [3]. The method proposed in [2] 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 kinematics quantities, which makes it particularly robust to uncertainties in parameter estimation and, additionally, computationally very effective. In general, torque redundancy is resolved by minimizing a cost criterion, e.g., a quadratic cost in the commands as in [2].

In the case of legged robots, it is desirable to use the torque redundancy to directly influence the ground reaction forces instead. Sentis et al. [4] showed, for example, how to directly control the generalized forces created at the contacts by manipulating the torques acting in the nullspace of the motion associated with the redundancy. Such an approach is interesting if one has a precise objective for the contact forces. However, it has the drawback of computational complexity and the inability to guarantee the achievement of the desired forces since it will depend on a nullspace projection.

Hyon et al. [5] proposed a force control approach to control the balance of a humanoid robot. The algorithm directly controls desired contact forces with the environment to achieve a desired task (e.g. a balance task). The advantage of such an approach is that it does not require a full dynamic model of the robot. However, the controller is derived assuming a static robot and therefore cannot take into account the robot motion. Stephens et al. [6] also proposed a force control approach for the balance of a humanoid robot. The method computes joint torques to achieve a desired contact forces objective using a dynamic model of the robot and constrained optimization algorithms. In both approaches [5], [6], the main goal of the controller is to maintain balance using force control while the motion of the robot can be viewed as a secondary task. In this contribution we take a different approach since our main objective is the realization of a desired acceleration for the robot and we use torque redundancy in a second phase to optimize contact constraints.

In [7], we showed how we could exploit torque redundancy to create contact constraints following Gauss’\(^1\) Principle of Least Constraints. However, in this preliminary result we

---

\(^1\)What we refer to the floating-base are the 6 DOFs needed to describe the position and orientation of the robot with respect to an inertial frame.

L. Righetti and S. Schaal, Computational Learning and Motor Control Lab, University of Southern California Los Angeles, CA90089, USA and Max Planck Institute for Intelligent Systems, Tübingen, Germany. ludovic.righetti@ai3.epfl.ch, sschaal@usc.edu
Jonas Buchli, Dept. of Advanced Robotics, Italian Institute of Technology, Genoa, Italy jonas@buchli.org
Michael Mistry, Disney Research Pittsburgh, Pittsburgh, PA15213, USA mmistry@disneyresearch.com

This research was supported in part by National Science Foundation grants ECS-0326095, IIS-0535282, IIS-1017134, CNS-0619937, IIS-0917318, CBET-0922784, EEC09-026052, 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.
didn’t know how to directly optimize an arbitrary quadratic cost in the contact constraints. Our controller was then not very useful for contact interaction of legged robots as we couldn’t directly take into account the geometry of the contacts to minimize, for example, the tangential forces. In this contribution we generalize our previous approach in order to minimize any quadratic cost in the contact constraints, using the controller developed by Mistry et al. [2]. Such a generalization is very important to make our approach relevant for legged robots. We show how we can use this general result to minimize the tangential contact forces during locomotion. We also show simulation experiments for a humanoid and a quadruped robot to demonstrate the improvement in locomotion gained with this new controller. We show that the robot is able to locomote on terrains with low friction with good performances where other controllers would fail. Finally we show an experiment on a real quadruped robot that demonstrates the robustness of the approach for real world applications.

II. PROBLEM FORMULATION

A. Rigid body dynamics model

Assuming a legged robot in contact with its environment with rigid body dynamics, its equations of motion are

\[ \dot{\mathbf{M}} \ddot{\mathbf{q}} + \mathbf{h} = \mathbf{S}^T \tau + \mathbf{J}_e^T \lambda \]

under the \( k \) contact constraints

\[ \mathbf{J}_e \ddot{\mathbf{q}} = \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) \]

where \( \mathbf{q} = [x_j^T \quad x_j^T]^T \) is the vector of joint positions \( (x_j \in \mathbb{R}^n) \) and base positions and orientations \( (x_b \in SE(3)) \), \( \mathbf{M} \in \mathbb{R}^{(n+6) \times (n+6)} \) is the rigid body dynamics inertia matrix, \( \mathbf{h} \in \mathbb{R}^{n+6} \) is a generalized force vector containing all forces (i.e. Coriolis and centrifugal forces, gravitation, friction, etc...), \( \tau \in \mathbb{R}^n \) is the actuation vector and \( \mathbf{S} \in \mathbb{R}^{n \times (n+6)} \) is the joint selection matrix. \( \mathbf{J}_e \in \mathbb{R}^{k \times (n+6)} \) is the Jacobian of the \( k \) constraints with the \( \lambda \in \mathbb{R}^k \) Lagrange multipliers that correspond to the constraint forces and \( \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^k \) specifies arbitrary constraints.

Following the ideas from [8] 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 \( \mathbf{J}_e \) 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 \( \mathbf{J}_e \).

For example, denoting the position of the foot of a humanoid robot by \( x_c \), then the constraint that the foot does not move can be written as \( x_c = \text{constant} \) or equivalently by \( x_c = 0 \). Relating this to the motion of the joints of the robot using the Jacobian of \( x_c \) we have \( \mathbf{J}_e \ddot{\mathbf{q}} = x_c = 0 \), which we differentiate once again to get \( \mathbf{J}_e \dddot{\mathbf{q}} = -\mathbf{J}_e \dot{\mathbf{q}} \).

Here we assume that the desired motion of the robot is given by desired joint accelerations and are constraint consistent. In other words, Equation (2) with \( \ddot{\mathbf{q}} = \dddot{\mathbf{q}}_d \) holds.

These accelerations will be satisfied if and only if they are of the form

\[ \dddot{\mathbf{q}}_d = \mathbf{J}_c^{G} \mathbf{b} + (\mathbf{I} - \mathbf{J}_c^{G} \mathbf{J}_c) \dddot{\mathbf{q}}_0 \]

where \( \mathbf{J}_c^{G} \) can be any generalized inverse [9] of \( \mathbf{J}_c \), i.e. a matrix such that \( \mathbf{J}_c \mathbf{J}_c^{G} \mathbf{J}_c = \mathbf{J}_c \). \( \dddot{\mathbf{q}}_0 \) is an arbitrary acceleration vector that is projected in the nullspace of the constraints thought \( (\mathbf{I} - \mathbf{J}_c^{G} \mathbf{J}_c) \).

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

- \( k < 6 \), the system is underactuated. There are more dimensions of motion 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 also need to be consistent with the dynamics of Eq. (1). It is the case, for example, 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 \( \dddot{\mathbf{q}}_d \). It is the case, for example, when a humanoid has both feet flat on the ground, or with one foot and one hand in contact with the environment or when a quadruped with point feet has more than two feet on the ground. It is the case of torque redundancy that is the topic of this paper.

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. In the other cases, it is not possible to optimize anything at the torque level, given a reference acceleration.

B. Inverse dynamics solution using orthogonal projections

As mentioned previously, [2] 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 they use the QR decomposition of the constraint Jacobian \( \mathbf{J}_c^T = \mathbf{Q} [\mathbf{R}^T \quad \mathbf{0}]^T \), where \( \mathbf{Q} \in \mathbb{R}^{(n+6) \times (n+6)} \) is an orthogonal matrix (i.e. \( \mathbf{QQ}^T = \mathbf{I} \)) and \( \mathbf{R} \in \mathbb{R}^{k \times k} \) is an upper triangular invertible matrix. If we decompose \( \mathbf{Q} = [Q_c \quad Q_u] \) into the constrained \( Q_c \in \mathbb{R}^{(n+6) \times k} \) and unconstrained \( Q_u \in \mathbb{R}^{(n+6) \times (n+6 - k)} \) components, the general solution for the inverse dynamics torques given desired accelerations \( \dddot{\mathbf{q}}_d \) can be written as

\[ \tau(W, \tau_0) = \frac{\mathbf{Q}_u^T \mathbf{S}^T \mathbf{Q}_u^T (M \dot{\dddot{\mathbf{q}}}_d + \mathbf{h})}{\mathbf{I} - \mathbf{Q}_u^T \mathbf{S}^T \mathbf{Q}_u \mathbf{S}^T \mathbf{W}} \mathbf{W}^{-1} \tau_0 \]

with generalized inverse

\[ \mathbf{Q}_u^T \mathbf{S}^T = \mathbf{W}^{-1} \mathbf{S} \mathbf{Q}_u (\mathbf{Q}_u^T \mathbf{S}^T \mathbf{W}^{-1} \mathbf{S} \mathbf{Q}_u)^{-1} \]
where $W \in \mathbb{R}^{n \times n}$ is any symmetric positive definite matrix and $\tau_0$ is an arbitrary internal torque. It is premultiplied by a projection matrix that guarantees that $\tau_0$ can only create internal forces, but no movement.

Moreover the generated constraint forces can be predicted by

$$\lambda = R^{-1}Q_c^T(Mq_d + h - S^T \tau)$$

(Remark 1: In the case where $k = 6$, there is only one solution and $Q_c^T S^T = (Q_c^T S^T)^+$ and the nullspace is empty. When $k < 6$, there is at most one solution and $Q_c^T S^T = (Q_c^T S^T)^+$, where $()^+$ denotes the Moore-Penrose generalized inverse and the nullspace is also empty.

We note that the control law $\tau(W, \tau_0)$ is parametrized by a weight matrix and an internal torque vector. This parametrization can be exploited to resolve torque redundancy in order to minimize some cost. One result that is already known from [2] and [10] is that $\tau_0 = 0$ leads to the minimization of the cost $\tau^T W \tau$ at each instant of time, which is similar to results from operational-space approaches for manipulators [11]. Torque redundancy can be resolved by minimizing a quadratic cost in the commands. We will show in the following how we can do the same for quadratic costs in the constraint forces.

III. 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 optimal solution presented in Eq. (4).

Another solution would be to minimize a quadratic cost in the constraint forces that would take into account those inequalities implicitly. If we can get the optimal constraint force distribution that minimizes such a cost, then, implicitly, it would be the most conservative possible solution that tries to fulfill the inequalities. If it is not possible, then the desired accelerations might need to be redesigned. The other 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.

A. General results

In the following lemma we present the first step towards the minimization of quadratic costs in the constraints. It is a more general formulation of our previous result in [7].

**Lemma 1:** Given a symmetric positive definite matrix $W_c$, the controller that minimizes the cost

$$\lambda^T J_c W_c J_c^T \lambda$$

at each instant of time and achieves the desired accelerations is selected by choosing

$$W = SW_c S^T$$

$$\tau_0 = SW_c (Mq_d + h)$$

Proof: We have a minimization problem, under the constraint that $\tau$ is a specific parametrization of the possible controllers, i.e. that the controller is coherent with the equations of motion. The constraint can then be written as

$$Q_c^T S^T \tau = Q_c^T (Mq + h)$$

The Lagrangian of the optimization problem is

$$L = \frac{1}{2} \lambda^T J_c W_c J_c^T \lambda + \alpha^T (Q_c^T S^T \tau - Q_c^T (Mq + h))$$

where $\alpha$ is a vector of Lagrange multipliers to be determined. Using the equation of motion Eq. (1) we can further develop it into

$$L = \frac{1}{2} (Mq + h - S^T \tau)^T W_c (Mq + h - S^T \tau) + \alpha^T (Q_c^T S^T \tau - Q_c^T (Mq + h))$$

Since this is a convex optimization problem, there is only one solution and this solution is determined by the condition

$$\nabla_{\tau, \alpha} L = 0$$

which can explicitly be written as the set of equations

$$SW_c S^T \tau = SW_c (Mq + h) - SQ_c \alpha$$

$$Q_c^T S^T \tau = Q_c^T (Mq + h)$$

From the first equation we find

$$\tau = (SW_c S^T)^{-1} (SW_c (Mq + h) - SQ_c \alpha)$$

Note that $SW_c S^T$ is invertible because $W_c$ is symmetric positive definite and therefore every upper left sub-matrix has a positive determinant and multiplication with $S$ selects one of these sub-matrices. We insert this in the second equation:

$$(SQ_c)^T (SW_c S^T)^{-1} (SW_c (Mq + h) - SQ_c \alpha) = Q_c^T (Mq + h)$$

which leads to

$$(SQ_c)^T (SW_c S^T)^{-1} SQ_c \alpha = ((SQ_c)^T (SW_c S^T)^{-1} SQ_c)(Mq + h)$$

which has for solution

$$\alpha = ((SQ_c)^T (SW_c S^T)^{-1} SQ_c)^+ \times ((SQ_c)^T (SW_c S^T)^{-1} SW_c - Q_c^T) (Mq + h)$$

then inserting this into the previous optimal torque formulation we get

$$\tau = (SW_c S^T)^{-1} \times (SW_c - SQ_c (Q_c^T S^T SW_c S^T)^{-1} SQ_c)^+ \times (Q_c^T S^T SW_c S^T)^{-1} SW_c - Q_c^T)(Mq + h)$$
which can be rearranged in the form

\[ \tau = \hat{Q}^T_u S^T Q^T_u (M\dot{q} + h) + (I - \hat{Q}^T_u S^T Q^T_u S^T) \tau_0 \]

with \( W = SW_cS^T \) and \( \tau_0 = SW_c(M\dot{q} + h) \), which completes the proof.

We now have a relation between redundancy resolution and optimality for constraint forces. However the form of the quadratic cost is not very intuitive since it involves the constraint Jacobian, which projects – in a non trivial way – the constraint forces. The following corollary solves this problem

Corollary 1: If \( W_\lambda \) is symmetric positive definite, a weight that minimizes the cost

\[ \lambda^T W_\lambda \lambda \]  

(10)

can be chosen to be

\[ W_c = Q \begin{bmatrix} R^{-T}W_\lambda R^{-1} & 0 & 0 \\ 0 & I \end{bmatrix} Q^T \]  

(11)

where we used the QR decomposition of the constraint Jacobian \( J_c^T = Q \begin{bmatrix} R & 0 \end{bmatrix} \).

Proof: If \( W_\lambda \) is symmetric positive definite then \( W_c \) is symmetric positive definite. Then straightforward calculations using the QR decomposition of \( J_c^T \) and the fact that \( Q \) is orthogonal (i.e. \( Q^T Q = I \)) shows that \( \lambda^T J_c W_c J_c^T \lambda = \lambda^T W_\lambda \lambda \) which finishes the proof.

This result is the main theoretical contribution of the paper and provides us with a way to directly and explicitly optimize constraint forces using torque redundancy. Now cost functions can be designed depending on the desired application in order to manipulate the generated constraint forces.

Remark 2: This result is very general as it can be applied for any type of constraints expressed in the form of Eq. (2) and for any quadratic cost of those constraints.

Remark 3: It must be noted that the symmetric positive definite matrix \( W_\lambda \) must be chosen such that the units in the cost are consistent if the constraint forces \( \lambda \) have different units (for example a mixture of forces and torques).

B. Minimization of tangential contact forces

Using the results of the previous section we can now create controllers with interesting properties for legged robots.

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 momentum 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 (Figure 1).

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

\[ W_\lambda = \begin{bmatrix} R_{leg_1}^T W_{leg_1} R_{leg_1} & 0 & \cdots \\ 0 & \cdots & R_{leg_n}^T W_{leg_n} R_{leg_n} \end{bmatrix} \]  

(12)

with

\[ W_{leg_i} = \text{diag}(K_{tx}, K_{ty}, 1, K_{mx}, K_{my}, K_{mz}) \]  

(13)

where \( R_{leg_n} \) 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 momenta of a foot with the orientation of the surface). \( K_i \) are gains associated with the tangential forces in x,y directions and the momenta in x,y,z directions. Here we assume that the z direction is normal to the ground.

Remark 4: In the case of point feet the matrix \( W_{leg} \) is reduced to its \( 3 \times 3 \) upper-left sub-matrix.

IV. Experimental results

In this section we present various experiments that illustrate the advantages of using the previously proposed redundancy resolution scheme and preliminary applications on real robots. In order to show its generality, we apply our approach to two different simulated robots (Figure 2) using different types of controllers and planners that generate the desired accelerations \( \dot{q}_d \).
A. Control of a humanoid robot

Our first results concern a simulation of the Sarcos humanoid robot (Fig. 2) which is a 34-DOFs human size humanoid robot. The goal of these experiments is to show how the distribution of reaction forces changes with our redundancy resolution scheme in a simple example, in order to illustrate the main properties of the controller.

In these simulations, the goal of control is to keep the robot at a desired posture while using our redundancy resolution scheme to minimize tangential ground reaction forces together with the moment generated around the feet. Comparisons are given with the original inverse dynamics controller that minimizes the total command cost.

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 3. 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. Such a controller will be even more beneficial in scenario involving more complex contact interactions, such as a humanoid taking support on a table with its arms while standing up from a chair.

B. 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\(^2\). The planning of desired joint positions, velocities and accelerations is done following the method proposed in [12]. We use a ZPM-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 \(q_d, \dot{q}_d\) and \(\ddot{q}_d\), is therefore

\[
\tau = \tau(W, \tau_0) + \text{PID}(q_d, \dot{q}_d)
\]

where \(\text{PID}\) corresponds to a joint space PID error feedback controller.

We use a physical simulation of the Little Dog robot (Fig. 2). 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 forces are shown in Figure 3. We notice that on the flat surface, both controllers generate similar contact forces and moments. 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. Such a controller will be even more beneficial in scenario involving more complex contact interactions, such as a humanoid taking support on a table with its arms while standing up from a chair.

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 command cost. For each experiment we set \(K_{tx} = K_{ty} = 1000N^{-2}\) and \(K_{mx} = K_{my} = K_{mz} = 2000Nm^{-2}\).

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 command cost. For each experiment we set \(K_{tx} = K_{ty} = 1000N^{-2}\) and \(K_{mx} = K_{my} = K_{mz} = 2000Nm^{-2}\).

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 slope. 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 Figure 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

---

\(^2\)These experiments are also shown in the supplemental video.
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.

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.

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. Also, the robot controlled by 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.

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.

C. Application to real robot

In this section we present preliminary 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 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 as our physical simulator. The controller was run at 3 different speeds. 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, for all the experiments the amount of slipping was significantly reduced by the controller with optimal distribution of contact forces. Figure 5 shows snapshots of a typical locomotion behavior for both tested controllers. We see that the new controller made the robot climb the slope faster due to the reduction in slipping. Figure 6 also shows the different amount in slipping for different experiments for both controllers. The amount of slip is computed accurately thanks to a motion capture
Fig. 5. Snapshot of a walking sequence of the LittleDog. Upper sequence is the controller with optimal distribution of contact forces, the lower sequence is the normal inverse dynamics controller. We notice that the robot climbs the slope faster with the optimal controller due to the significant reduction in slipping. The red circles show that the feet of both robots are synchronously touching the ground.

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

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 controller we proposed in this paper.

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 clear behavioral differences. However a few limitations of the robotic platform can explain the quality of the results:

• As we discussed previously, the inverse dynamics controller is running at a 100Hz bandwidth on a host computer. Such a low bandwidth of control clearly limits the performance of the robot. For example, on a modern torque controlled humanoid 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 also plays 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.

V. CONCLUSION

In this paper, we addressed the problem of torque redundancy for inverse dynamics control of floating-base robots under contact constraints. Our main theoretical result is the derivation of a redundancy resolution scheme that minimizes any quadratic cost in the contact constraints. We then proposed to use this result to minimize tangential contact forces when controlling legged robots. 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.

Extensive simulation results showed that, given desired trajectories, we could 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 preliminary 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. Our research also emphasizes the need for an appropriate robot design that can exploit advanced torque control schemes.

REFERENCES


