

Robustness to Joint-Torque-Tracking Errors in Task-Space Inverse Dynamics

Andrea del Prete, Nicolas Mansard

▶ **To cite this version:**

Andrea del Prete, Nicolas Mansard. Robustness to Joint-Torque-Tracking Errors in Task-Space Inverse Dynamics. IEEE Transactions on Robotics, 2016, 32 (5), pp. 1091-1105. 10.1109/TRO.2016.2593027 . hal-01241974v2

HAL Id: hal-01241974
https://hal.science/hal-01241974v2

Submitted on 31 Aug 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - ShareAlike 4.0 International License

Robustness to Joint-Torque Tracking Errors in Task-Space Inverse Dynamics

Andrea Del Prete, *Member, IEEE,* and Nicolas Mansard

***Abstract*—Task-Space Inverse Dynamics (TSID) is a well-known optimization-based technique for the control of highly-redundant mechanical systems, such as humanoid robots. One of its main flaws is that it does not take into account any of the uncertainties affecting these systems: poor torque tracking, sensor noises, delays and model uncertainties. As a consequence, the resulting control-state trajectories may be feasible for the ideal system, but not for the real one. We propose to improve the robustness of TSID by modeling uncertainties in the joint torques, either as Gaussian random variables or as bounded deterministic variables. Then we try to immunize the constraints of the system to any—or at least most—of the realizations of these uncertainties. When the resulting optimization problem is too computationally expensive for online control, we propose ways to approximate it that lead to computation times below 1 ms. Extensive simulations in a realistic environment show that the proposed robust controllers greatly outperform the classic one, even when other unmodeled uncertainties affect the system (e.g. errors in the inertial parameters, delays in the velocity estimates).**

***Index Terms*—Legged Robots, Dynamics, Robust Control, Robust Optimization.**

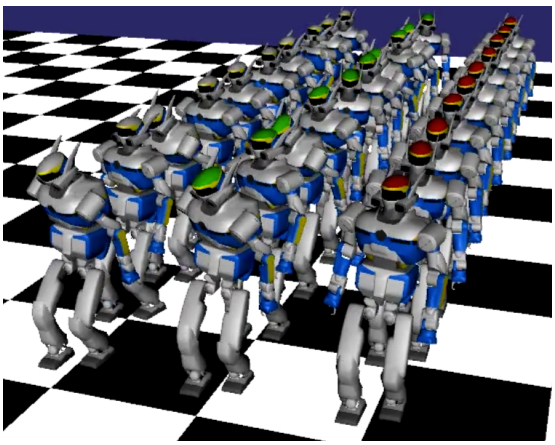


Fig. 1. Simulation of 30 HRP-2 robots walking in the presence of uncertainties, the goal being to compare the classic TSID controller (left line, gray heads) to the proposed robust TSID controllers: stochastic (central line, green heads) and worst-case (right line, red heads). Some of the simulation results can be seen in the accompanying video.

I. Introduction

TASK-SPACE inverse dynamics (TSID) has become an increasingly popular way to control humanoid and quadruped robots [1]–[5]. This success is motivated by two attractive features. First of all, it is theoretically sound [6] because it is based on an exact inversion of the dynamics of the system, resulting (in theory) in a perfect tracking of the desired trajectories—as long as they are feasible. Second, TSID is able to explicitly take into account bounds on the state and the actuation of the system, namely joint torques, accelerations and contact forces [7]. Thanks to these features TSID works extremely well in simulation [8], [9], which is why it has been used also in the graphics community to synthesize motion online [10], [11]. Another key factor is its computational efficiency [12], which allows its application for online control of real robots.

However, as usual, the gap between simulation and real world is large and can be explained through countless unmodeled uncertainties affecting these systems, such as poor torque control, model uncertainties, sensor noises and delays. This results in control trajectories that are feasible for the ideal system, but not for the real one. The recent results of the DARPA Robotics Challenge Finals [13] have shown promise for the range of tasks that can be accomplished in these frameworks [14]–[16], but have highlighted the critical need to address robustness concerns for these methods in order to prevent falls.

To improve the robustness of TSID-based control, we propose to account for such uncertainties, modeled as additive noise. Modeling uncertainties as random variables provides a generic framework that we can apply to the parameters of the robot model (e.g. inertias can be modeled as Gaussian distributions), the sensor measurements (e.g. additive noise on the measured velocity) or control inputs. Additionally, the noise distributions can be identified by statistical analysis. In particular, we focus on the case where uncertainties only affect the decision variables, i.e. the joint torques. The accuracy of the torque tracking is known to be an important issue [17], [18], in particular for robots that do not have access to a direct measurement of the joint-torques—such as most current humanoid robots: HRP-2, Hubo, Atlas, Valkyrie, Asimo, iCub. Additionally, robustness to the noise in the joint torques ensures also some level of robustness to several other uncertainties, like measurement delays and model inaccuracies, as shown by our simulations. Focusing on this class of uncertainties is also interesting because it leads to convex optimization problems, which are in general easier (and faster) to solve than nonconvex problems. Other types of noise that may be equally significant, such as measurement noise (especially velocity) or modeling errors, may be addressed in future work by extensions of the proposed methodology.

First, Section II introduces the issues arising from solving

The authors are with the CNRS, LAAS, 7 avenue du colonel Roche, Univ de Toulouse, LAAS, F-31400 Toulouse, France. e-mail: adelpret@laas.fr, nmansard@laas.fr.

Manuscript received December 9, 2015.

an optimization problem without accounting for uncertainties. Then Section II-A and II-B discuss the two main methods to account for uncertainties in optimization problems: stochastic [19] and deterministic [20]. In particular, we focus on the case in which uncertainties only affect the problem variables (i.e. the joint torques). If the uncertainties are stochastic, that is they are random variables following a known probability distribution, this results in a stochastic optimization problem. This problem is too computationally expensive to be used for online control (e.g. several seconds of computation for 30 variables and 90 inequality constraints). In Sections II-C and II-D, we discuss two ways to approximate a general stochastic optimization problem with linear inequality constraints. These approximations greatly reduce the computation time, while maintaining a sufficiently-good accuracy. Section III shows how these ideas relate to the TSID control problem. We then discuss in more details the ideas presented in Section II to get different formulations of *robust TSID* (Sections IV and V). In Section VI we validate the proposed methods on a simulated HRP-2 humanoid robot performing walking (see Fig. 1) and manipulation tasks. Through extensive simulations under realistic conditions (i.e. uncertainties in the joint torques, joint velocities and inertial parameters) we empirically show that taking robustness into account greatly increases the chances of the robot not to fall. Moreover, we verify that we can solve the proposed optimization problems in less than 1 ms on a standard CPU, so that these formulations are suitable for online control. Finally, Section VII presents the related works and Section VIII summarizes the paper before discussing the future work.

With respect to our previous work [21] this paper presents several new contributions.

- In Section IV we discuss the possibility of modeling noise as a bounded variable rather than as a random variable, resulting in a worst-case optimization.
- In Section V-A1 we show an interesting connection between the proposed stochastic optimization and the log-barrier methods ([22], ch. 17) used to solve inequality-constrained problems.
- In Section VI-B we present data collected on a real torque-controlled robot that validate our modeling assumption for the joint-torque tracking errors.
- In Section VI-F and VI-G we present new *extensive* simulation results to compare the proposed robust controllers to the classic one in a *realistic* simulation environment. We selected a walking task and a drilling task, for which we present statistics based on several batches of 100 tests each; in each batch we simulated different uncertainties, in terms of type (torque bandwidth, torque noise, velocity delays, inertial parameter errors) and magnitude.

II. Introducing Uncertainties

Before going into the details of our approach, this section overviews the key ideas that we propose to use to make TSID robust. The problem of controlling a robot with TSID

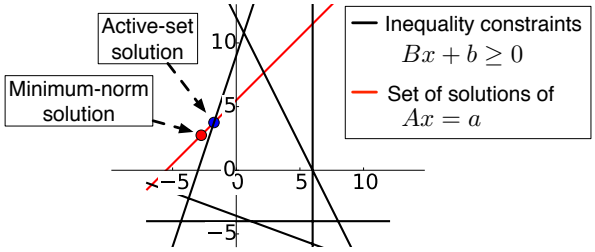


Fig. 2. 2D example of an inequality-constrained least-square problem, such as (1), solved by an active-set method.

can be cast as the following abstract constrained least-square optimization with n variables and m inequality constraints:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & ||Ax - a||^2 \\ \text{subject to} \quad & Bx + b \geq 0 \end{aligned} \tag{1}$$

In Section III we will present the exact formulation of TSID that we use, in which x represents the joint torques. Problem (1) may have infinitely many optima (in case A does not have full column rank) so the determined solution may depend on the technique that we use to solve it. The most common approach to solve problem (1) in robotics is through active-set methods [12], mainly because they are easy to warm-start[1] (contrary to interior-point methods [23]). Because of their working principle, active-set algorithms tend to find solutions that satisfy some inequality constraints with zero margin, which are poor in terms of robustness.

Let us quickly look at how an active-set method works by using the 2D example depicted in Fig. 2. Different variants of the active-set algorithm exist; in this example we look at the most classical dual method [24], which starts its search at the unconstrained minimum of the objective function (i.e. red dot). Since this point violates an inequality constraint, it adds this constraint to the so-called *active set*, which is the set of constraints that are satisfied as equalities at the optimum. The new solution is represented by the blue dot. Clearly this solution has little robustness because infinitely-small changes in x, B or b could lead to violations of the active inequality constraint. Intuition suggests that we could instead choose a solution that has a higher chance to satisfy the inequalities by moving towards the internal part of the feasible solution space.

However, we do not want the reader to think that active-set algorithms are the only cause of poor robustness in TSID. Surely, if the optimum of the problem is not unique (as in this example) using interior-point methods would improve robustness. However, in general this would not be enough because i) it does not allow to sacrifice performance to improve robustness (which sometimes is necessary), and ii) often in TSID the optimum is unique. The real issue is the lack of a robustness measurement in the cost function, which would make the solution robust regardless of the used optimization algorithm.

[1]Warm-starting an optimization algorithm consists in exploiting the solution of a similar problem (which was already computed, typically at the previous control cycle) to speed-up the computation.

To come up with a measurement of robustness we first need to model the uncertainties in the TSID optimization problem. We then could resort to robust optimization techniques to find control inputs that are robust to these uncertainties. We can model uncertainties as deterministic variables belonging to a known set, and try to find a solution that is feasible for any realization of the uncertainty in the given set [25]. Alternatively, we can model uncertainties as random variables following known probability distributions, and try to find a solution that satisfies the constraints with a large-enough probability. People refer to the first case as *robust* or *worst-case optimization*, while the second case is known as *stochastic optimization* or *chance-constrained programming*. In the following we are going to explore both directions.

A. Deterministic Uncertainties

The tractability of a robust optimization problem is strictly connected to the geometric shape of its uncertainty set U. Since it is reasonable to assume that the torque tracking errors at different joints are independent from each other, using an hyper-rectangle as uncertainty set seems a good choice:

$$e \in U, \quad U = \{z \in \mathbb{R}^n : |z| \leq e^{max}\},$$

where $e \in \mathbb{R}^n$ is the torque tracking error, $e^{max} \in \mathbb{R}^n$ is its maximum value and $|z|$ is a vector containing the absolute value of the elements of z. The robust optimization problem is then:

$$\underset{x}{\text{minimize}} \quad ||Ax - a||^2 \tag{2a}$$

$$\text{subject to} \quad B(x + e) + b \geq 0 \quad \forall e \in U \tag{2b}$$

We did not model the uncertainties in the cost function to avoid having a too conservative behavior of the system, which is a well-known issue in worst-case optimization [26]. The problem is not tractable in this form because the constraint (2b) actually hides an infinite number of constraints. Another issue that we need to address is the potential infeasibility of this robust problem: there may not exist a value of x that satisfies the constraints for any realization of e. In this case we need to relax the robust constraints so as to find a solution that satisfies (at least) the standard constraints. In Section IV we reformulate (2) as a standard QP that ensures the feasibility of the TSID problem.

B. Stochastic Uncertainties

Alternatively, we can model uncertainties as Gaussian noise $e \sim \mathcal{N}(0, \Sigma)$ with a decoupled covariance matrix $\Sigma = \text{diag}(\begin{bmatrix}\sigma_1^2 & \dots & \sigma_n^2\end{bmatrix})$ affecting the decision variable x:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & ||A(x + e) - a||^2 \\ \text{subject to} \quad & B(x + e) + b \geq 0 \end{aligned} \tag{3}$$

Since e is a random variable, both cost and constraints are now random variables, so (3) does not make sense. Rather than minimizing the cost function we can minimize its expected value, but since e has zero mean, this actually does not change the problem:

$$\mathbf{E}||A(x + e) - a||^2 = ||Ax - a||^2 + \text{Tr}(A^\top A\Sigma)$$

The inequalities are less trivial and consequently less frequently considered. The classic approach in chance-constrained programming is to replace them with their probability to be satisfied [19]:

$$p(x) = \text{P}(B(x + e) + b \geq 0) \tag{4}$$

In general $p(.)$ is not convex, so it is not wise to use it directly in our optimization problem. A better approach is to define a convex function $R(.)$ that is monotonically decreasing with respect to $p(.)$[2], and then insert it in the cost function to find a trade-off between performance (i.e. small cost) and robustness:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & ||Ax - a||^2 + wR(x) \\ \text{subject to} \quad & Bx + b \geq 0, \end{aligned} \tag{5}$$

where $w \in \mathbb{R}$ weighs the importance of robustness with respect to cost. Here, we kept the deterministic inequalities to prevent the solution from violating them in favor of minimizing the cost (which may happen if w is not large enough).

Alternatively, rather than looking for a trade off, we could apply a strict prioritization approach [6], [27]–[29]. In other words, we could either maximize robustness in the null space of the cost, or minimize the cost subject to the constraint of $R(x)$ being greater than a certain value. Even if these prioritization approaches are interesting, they would require solving two optimization problems in cascade, unless a dedicated lexicographic solver is available [12]. This would increase the total computation time, so we decided not to use them.

To solve (5) we need to evaluate the cumulative density function (CDF) of the multivariate random variable $e_B = Be$, that is $\text{P}(e_B > -b - Bx)$. In general there is no analytical expression to compute this CDF, and resorting to numerical techniques [30] would make the computation too slow for applications in control (e.g. about 0.5 s for 90 inequalities and 30 variables). Aside from introducing robustness in TSID, the main contribution of this paper is to propose two approximations of (4) that are much faster to compute and that provide satisfying precision and robustness in practice.

C. Approximation 1—Individual Constraints p_{ind}

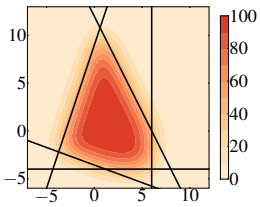
The first way to simplify (4) is by considering the probabilities of the single inequalities rather than the probability of all of them:

$$p_{ind}(x) = \prod_{i=1}^{m} \text{P}(B_i(x + e) + b_i \geq 0), \tag{6}$$

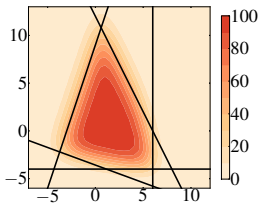
where B_i is the i-th row of B. When e is Gaussian, this is equivalent to neglecting the off-diagonal terms of the covariance matrix of e_B. Thanks to this approximation we can solve (4) much faster because we only need m univariate CDFs—rather than one multivariate CDF.

To get an intuition of why $p_{ind}(x)$ is a good approximation of $p(x)$ let us look at a simple 2D example (we will see how the proposed probability approximations perform in a high-dimensional space in Section VI-C). Fig. 3a depicts the probability $p(x)$ to satisfy a set of 5 linear stochastic inequalities,

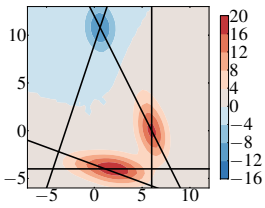
[2]For instance, rather than maximizing a probability we minimize its negative logarithm.



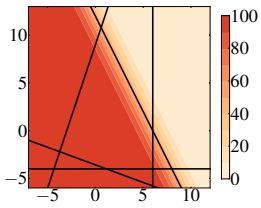
(a) Joint inequalities probability $p(x) = \mathrm{P}(B(x+e)+b > 0)$.



(b) Individual inequalities probability $p_{ind}(x)$.

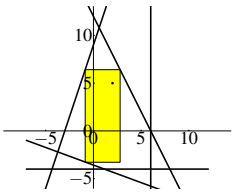


(c) Difference between Fig. 3a and 3b: $p(x) - p_{ind}(x)$; mean error 1.8%.

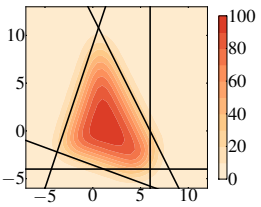


(d) Probability of a single inequality constraint $\mathrm{P}(B_0(x+e)+b_0 \geq 0)$.

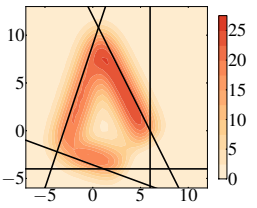
Fig. 3. 2D example comparing $p(x)$, namely the probability of satisfying a set of affine stochastic inequalities, with its approximation $p_{ind}(x)$ (6).



(a) Hyper-rectangle approximation.



(b) Hyper-rectangle probability p_{box}.



(c) Hyper-rectangle probability error: $p(x) - p_{box}(x)$; mean error 4.7%.

Fig. 4. 2D example comparing $p(x)$ with its approximation $p_{box}(x)$ (8).

with e having a standard deviation $\sigma_1 = \sigma_2 = 1.44$. Fig. 3b shows the approximated probability $p_{ind}(x)$ obtained with (6), while Fig. 3d shows the probability of a single inequality. The overall shapes of the approximated and the real probability are quite similar and it is hard to spot the differences. To highlight the errors Fig. 3c shows the difference between $p(x)$ and $p_{ind}(x)$. The errors are concentrated at the intersections of the inequalities: when the angle between the inequalities is less than $90°$ the error is negative, when the angle is greater than $90°$ the error is positive, whereas when the angle is exactly $90°$ the error is void. The fact that the differences are concentrated at the intersections of multiple inequalities is actually advantageous: these regions have little robustness, so we can expect that, most of the time, our solution should not be there. Note that despite the tendency of underestimating risk close to corners with acute angles, solutions are located closer to the central part of the feasible region, where the risk is even lower. Moreover we can expect this approximation to work well as long as there are few constraints that are active at the same time, which is typically the case in TSID.

D. Approximation 2—Largest-Enclosed Hyper-Rectangle p_{box}

In the first approximation we exploited the fact that it is easy to compute the probability of a single inequality. Another case in which we can easily compute the probability is when all the inequalities are simple bounds (i.e. they define a hyper-rectangle aligned with the main axes). In this case, the joint probability is the product of n probabilities of univariate random variables, i.e. there is no more coupling. Our idea is then to approximate the real polyhedra with a hyper-rectangle $U(s)$ (where s is a parametrization of the hyper-rectangle) that is enclosed in it:

$$p(x) \approx \mathrm{P}((x+e) \in U(s)) \tag{7}$$

Of course for any $U(s)$ enclosed in the feasible set, the probability to be in $U(s)$ is lower than the probability to be in the feasible set $p(x)$. It follows that, among all the enclosed hyper-rectangles, the one resulting in the best approximation of $p(x)$ is the one that maximizes $\mathrm{P}((x+e) \in U(s))$:

$$\begin{aligned} p_{box}(x) = \underset{s}{\text{maximize}} \quad & \mathrm{P}((x+e) \in U(s)) \\ \text{subject to} \quad & Bz + b \geq 0 \qquad \forall z \in U(s), \end{aligned} \tag{8}$$

where the (infinitely many) constraints ensure that $U(s)$ is enclosed in the feasible set. Fig. 4a shows the hyper-rectangle maximizing the probability. Fig. 4b shows the value of p_{box} over the solution space, whereas Fig. 4c shows the approximation error $(p(x) - p_{box}(x))$. While this approximation may seem much coarser than the first one, in Section VI we will show empirically that it performs well in practice. Moreover, in Section V-B we will prove that the adopted single-variable parametrization of the hyper-rectangle results in a linear optimization problem, which is easier to solve than the nonlinear problem resulting from our first approximation.

III. Robust Task-Space Inverse Dynamics

Various formulations of the TSID optimization problem exist and are often equivalent or similar [6]. We write it here as an optimization problem of $x = (\dot{v}, f, \tau)$ [31]:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & ||Ax - a||^2 \\ \text{subject to} \quad & Bx + b \geq 0 \\ & \begin{bmatrix} J_c & 0 & 0 \\ M & -J_c^\top & -S^\top \end{bmatrix} \begin{bmatrix} \dot{v} \\ f \\ \tau \end{bmatrix} = \begin{bmatrix} -\dot{J}_c v \\ -h \end{bmatrix}, \end{aligned} \tag{9}$$

where $\dot{v} \in \mathbb{R}^{n+6}$ are the base and joint accelerations, $f \in \mathbb{R}^k$ are the contact forces, $\tau \in \mathbb{R}^n$ are the joint torques, $J_c \in \mathbb{R}^{k \times (n+6)}$ is the constraint Jacobian, $M \in \mathbb{R}^{(n+6)\times(n+6)}$

is the mass matrix, $h \in \mathbb{R}^{n+6}$ contains the bias forces and $S \in \mathbb{R}^{n\times(n+6)}$ is the selection matrix. The inequality constraints (defined by B and b) can represent the torque limits, the (linearized) force friction cones, the ZMP bounds and the joint-acceleration limits. The bounds of the joint positions and velocities are typically converted into joint-acceleration bounds [32]. The cost function represents the error of the task, which is typically an affine function of $\dot{v}$ (i.e. a task-space acceleration):

$$\underbrace{\begin{bmatrix} J_{task} & 0 & 0 \end{bmatrix}}_{A} x - \underbrace{(\ddot{x}^{des}_{task} - \dot{J}_{task} v)}_{a} = \ddot{x}_{task} - \ddot{x}^{des}_{task}$$

The task may be to track a predefined trajectory of a link, of the center of mass of the robot, or to regulate the robot's angular momentum.

This problem is rather similar to the one that we considered in the previous section, apart from the fact that it has equality constraints. Without knowing the value taken by the uncertainty $e \in \mathbb{R}^n$ in the joint torques we can not select a value of x that satisfies the equality constraints. For this reason, we reformulate (9) with respect to τ alone by expressing $\dot{v}$ and f as functions of τ [3]:

$$\begin{bmatrix} \dot{v} \\ f \\ \tau \end{bmatrix} = \underbrace{\begin{bmatrix} M^{-1}N_c^\top S^\top \\ \Lambda_c J_c M^{-1} S^\top \\ I \end{bmatrix}}_{C} \tau + \underbrace{\begin{bmatrix} -M^{-1}(N_c^\top h + J_c^\top \Lambda_c \dot{J}_c v) \\ \Lambda_c (J_c M^{-1} h - \dot{J}_c v) \\ 0 \end{bmatrix}}_{c},$$

where $\Lambda_c = (J_c M^{-1} J_c)^{-1}$ and $N_c = I - M^{-1} J_c^\top \Lambda_c J_c$. Then the problem takes on the following form:

$$\begin{aligned} \underset{\tau}{\text{minimize}} \quad & ||D\tau - d||^2 \\ \text{subject to} \quad & G\tau + g \geq 0, \end{aligned} \tag{10}$$

where $D = AC$, $d = a - Ac$, $G = BC$, $g = Bc + b$. Note that (10) is equivalent to (9). Even if (10) has only τ as decision variable, it can contain constraints and costs related to $\dot{v}$ and f. Now that we cast TSID in the same form as (1), we can use the ideas presented in the previous section to introduce robustness in this problem.

IV. Deterministic Uncertainties

Following the idea of Section II-A the robust TSID optimization problem is:

$$\begin{aligned} \underset{\tau}{\text{minimize}} \quad & ||D\tau - d||^2 \\ \text{subject to} \quad & G(\tau + e) + g \geq 0 \quad \forall e \in U \end{aligned} \tag{11}$$

Now we will show how to get rid of the infinite constraints of this problem to reformulate it in standard form; then we will explain how to deal with the cases in which the problem is infeasible.

[3]In this paper we assume that J_c is full row rank, but these results can be extended to the case of J_c being rank deficient [33].

1) Reduction of the Infinite Number of Constraints: We can represent the infinite constraints of (11) as a finite number of constraints:

$$l_i(\tau) \geq 0 \qquad i = 1 \dots m,$$

where l_i is the solution of an optimization problem:

$$\begin{aligned} l_i(\tau) = \underset{e}{\text{minimize}} \quad & G_i(\tau + e) + g_i \\ \text{subject to} \quad & |e| \leq e^{max}, \end{aligned}$$

with G_i being the i-th row of G. In simple terms, we are saying that if (and only if) an inequality is satisfied for the minimum value of its left-hand side (over all of the possible uncertainties), then it is satisfied for all of the possible uncertainties. Thanks to the simple shape of U that we selected, this is a Linear Program with solution:

$$l_i(\tau) = G_i \tau - |G_i| e^{max} + g_i$$

The rationale behind this simplification is that we do not check that an inequality is satisfied for all the values of U: we only verify that it is satisfied for its *worst corner*. The worst corner is the one that will eventually collide with the hyper-plane defined by the inequality if you enlarge the hyper-rectangle. This allows us to reformulate (11) as a standard QP:

$$\begin{aligned} \underset{\tau}{\text{minimize}} \quad & ||D\tau - d||^2 \\ \text{subject to} \quad & G\tau - |G| e^{max} + g \geq 0, \end{aligned}$$

where $|G|$ is a matrix containing the absolute values of the elements of G.

2) Infeasibility: As we already mentioned, this problem may be infeasible. In this case we would like to have a solution that at least satisfies the standard inequality constraints, and if possible guarantees some level of robustness. We can achieve this by introducing a slack variable $s \in \mathbb{R}$ that allows the solver to continuously pass from the robust constraints to the classic ones:

$$\begin{aligned} \underset{\tau, s}{\text{minimize}} \quad & ||D\tau - d||^2 - ws \\ \text{subject to} \quad & G\tau - |G| e^{max} s + g \geq 0 \\ & 0 \leq s \leq 1, \end{aligned} \tag{12}$$

where $w \in \mathbb{R}$ is a large-enough value (e.g. 10^6). Whenever possible the solver will try to set $s = 1$, which results in the satisfaction of the robust constraints. When the robust constraints are not feasible, the solver will decrease s of the minimum amount necessary to make the constraints feasible. Only when necessary, the solver will set $s = 0$, which results in the satisfaction of the standard constraints only. In case we are not sure whether the standard constraints are feasible we can even allow s to take negative values.

V. Stochastic Uncertainties

In case of stochastic uncertainties we measure the robustness of a solution of (10) by some approximations of:

$$R(\tau) = -\log p(\tau) = -\log \mathrm{P}(G(\tau + e) + g \geq 0)$$

The robust optimization problem is then:

$$\begin{aligned} \underset{\tau}{\text{minimize}} \quad & ||D\tau - d||^2 + wR(\tau) \\ \text{subject to} \quad & G\tau + g \geq 0, \end{aligned} \tag{13}$$

where $w \in \mathbb{R}$ weighs the importance of robustness with respect to performance.

A. Approximation 1—Individual Constraints p_{ind}

Our first idea to approximate $p(x)$ is to consider the constraints individually:

$$p(\tau) \approx p_{ind}(\tau) = \prod_{i=1}^{m} \mathrm{P}(G_i(\tau + e) + g_i \geq 0)$$

While most distributions have an analytical expression to compute the CDF in the univariate case, the Gaussian distribution does not. However, expressions exist to approximate it with high accuracy and low computational cost [34] (e.g. polynomials). To compute p_{ind} we need to evaluate $\mathrm{P}(G_i(\tau + e) + g_i \geq 0)$. Since e is Gaussian, we have $e_{G_i} = G_i e \sim \mathcal{N}(0, \sigma_{G_i})$, where $\sigma_{G_i} = \sigma_i^2 G_i G_i^\top$. Hence:

$$\begin{aligned} &\mathrm{P}(G_i(\tau + e) + g_i \geq 0) = \mathrm{P}(e_{G_i} \geq -G_i\tau - g_i) = \\ &= \mathrm{P}(e_{G_i} \leq G_i\tau + g_i) = F_{G_i}(G_i\tau + g_i), \end{aligned}$$

where $F_{G_i}(.)$ is the CDF of e_{G_i}. We then define the robustness function as:

$$R_{ind}(\tau) = -\log p_{ind}(\tau) = -\sum_{i=1}^{m} \log F_{G_i}(G_i\tau + g_i)$$

This function is convex and twice differentiable, so we can easily minimize it using any variant of Newton's method [35] (see [21] for the expressions of gradient and Hessian of R_{ind}). The final robust TSID problem is then a convex optimization:

$$\begin{aligned} \underset{\tau}{\text{minimize}} \quad & ||D\tau - d||^2 - w\sum_{i=1}^{m} \log F_{G_i}(G_i\tau + g_i) \\ \text{subject to} \quad & G\tau + g \geq 0 \end{aligned} \tag{14}$$

1) Relationship with Log-Barrier Method: Looking at (14) one may notice a certain similarity to the *log-barrier method* ([22], ch. 17). This interior-point method is a technique to solve inequality-constrained optimization problems by removing the inequality constraints and injecting their logarithm in the cost function. For instance, the standard TSID problem (10) can be solved through a sequence of unconstrained problems with decreasing values of the parameter α:

$$\underset{\tau}{\text{minimize}} \quad ||D\tau - d||^2 - \alpha\sum_{i=1}^{m} \log(G_i\tau + g_i) \tag{15}$$

As α approaches zero, the solution of (15) approaches the solution of (10). The main difference between (14) and (15) is that the logarithmic barrier in (15) tends to infinity as one of the inequalities tends to zero, whereas this is not the case for (14) (see Fig. 5). This is the reason why we need to have inequality constraints in (14) to avoid violating them.

In the special case where $F_{G_i}(G_i\tau + g_i) \propto G_i\tau + g_i$ then the two optimization problems are equivalent. This happens if the uncertainty affects g (rather than τ) and if the probability distribution of the uncertainty is uniform (rather than Gaussian) and bounded above by zero (i.e. can only be nonpositive). We can then interpret log-barrier methods as stochastic optimization techniques that take into account additive nonpositive uniformly-distributed random uncertainties on the inequality constraints.

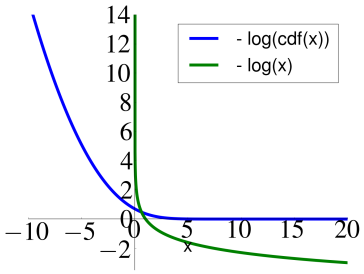


Fig. 5. Comparison of the logarithmic barrier used in (15) with the barrier used in (14). For the plot we considered a standard deviation of 2.0 for the CDF.

B. Approximation 2—Largest-Enclosed Hyper-Rectangle p_{box}

Our second idea is to approximate the polytope defined by the inequalities with a hyper-rectangle. We can compute this approximation by solving this optimization problem:

$$\begin{aligned} p_{box}(\tau) = \underset{s}{\text{maximize}} \quad & \mathrm{P}(e \in U(s)) \\ \text{subject to} \quad & G(\tau + z) + g \geq 0, \quad \forall z \in U(s) \end{aligned} \tag{16}$$

We parametrize $U(s)$ with a single variable[4] $s \in \mathbb{R}$:

$$U(s) = \{z \in \mathbb{R}^n : |z_i| \leq k_i s \quad i = 1 \ldots n\},$$

where $k \in \mathbb{R}^n$ encodes the fixed ratio between the n sides of the hyper-rectangle. Contrary to s, k is fixed and given by the user, so it is not a variable of the optimization. In our tests we have always set $k = (\sigma_1, \ldots, \sigma_n)$. A poor choice of k (due to a poor estimation of Σ) can clearly degrade the quality of the p_{box} approximation. However, estimating Σ on a real robot should be rather straightforward, so this should not be a problem in practice. Thanks to the fact that $\mathrm{P}(e \in U(s))$ is a monotonically increasing function of s, maximizing the probability over the hyper-rectangle is equivalent to maximizing s. Rather than solving (16) we can then solve:

$$\begin{aligned} R_{box}(\tau) = \underset{s \geq 0}{\text{maximize}} \quad & s \\ \text{subject to} \quad & G(\tau + z) + g \geq 0, \quad \forall z \in U(s) \end{aligned} \tag{17}$$

Note that in general $R_{box}(\tau) \neq p_{box}(\tau)$, but (16) and (17) result in the same value of s. Despite this simplification, using

[4]In our previous work [21] we also investigated the case of $U(s)$ being a general hyper-rectangle (aligned with the main axes), parametrizing it with $2n$ variables. However, the resulting optimization problem is harder to solve (it is nonlinear, sparse, and it has $3n$ variables) and the resulting controller performed worse than the controller using the p_{ind} approximation, so we decided to stop using it.

$R_{box}(\tau)$ in place of $R(\tau)$ may seem rather complex. First, we need to optimize a function (R_{box}) that is itself the solution of an optimization problem and second, (17) cannot be solved in this form since it has an infinite number of constraints. Despite appearances, this boils down to solving a simple QP.

Using the same method discussed in Section IV we can reformulate (17) as a linear problem whose solution is our robustness measure:

$$\begin{aligned} R_{box}(\tau) = \underset{s \geq 0}{\text{minimize}} \quad & -s \\ \text{subject to} \quad & G\tau - |G|ks + g \geq 0 \end{aligned} \tag{18}$$

Now that we got rid of the infinite number of constraints, we need to understand how we can minimize $R_{box}(\tau)$ with respect to τ, $R_{box}(\tau)$ being the solution of an optimization problem. The answer is surprisingly simple: we perform both optimizations at the same time, which gives us:

$$\begin{aligned} \underset{\tau, s \geq 0}{\text{minimize}} \quad & \frac{1}{2}||D\tau - d||^2 - ws \\ \text{subject to} \quad & G\tau - |G|ks + g \geq 0 \end{aligned} \tag{19}$$

This means that we look at the same time for the solution of the original problem τ and for the "best" enclosed hyper-rectangle. To solve this problem we can use a standard QP solver, as is usually the case for the classic TSID.

It is interesting to note that for $k = e^{max}$ (19) is almost identical to (12), that is the problem we got starting from a deterministic uncertainty set. The only difference is that in (12) the variable s cannot be larger than 1, whereas in (19) it has no upper bound. This is because the deterministic uncertainty is bounded, so it makes no sense to immunize the solution to uncertainties bigger than e^{max}. The stochastic uncertainty is instead unbounded, so the bigger s, the higher the robustness. Despite their resemblance, the parameter w has a different meaning in the two problems: in (12) it is just a large number that is supposed to approximate a strict priority of robustness with respect to performance, whereas in (19) it specifies the trade-off between robustness and performance. It is not advisable to set w to a too-large value in (19) because it would result in extremely poor tracking performance.

VI. Simulations

In this section, we present a series of simulation results that try to answer to the following questions:

- What improvement can we get in terms of probability to satisfy the inequalities by using robust TSID?
- Which of the proposed formulations performs better?
- Can we solve these optimization problems in under 1 ms?
- What would be the practical benefits of using a robust controller in a realistic scenario with unmodeled uncertainties, such as errors in the inertial parameters and in the estimated velocities?

We tested the proposed controllers on three typical humanoid tasks (whole-body reaching, whole-body manipulation, and walking) with the 30-degree-of-freedom humanoid robot HRP-2. Table I lists all the simulation parameters. As already discussed at the end of Section V-B, the deterministic robust controller is almost equivalent to the stochastic robust controller with p_{box} approximation. In particular, if we set $k_i = e_i^{max} = \alpha\sigma_i$ (for any $\alpha > 0$), the two controllers find exactly the same control action any time the solution of the p_{box} controller contains a value of $s \leq 1$. We verified that in our experimental conditions this happened almost always (for $\alpha = 3$), so we tested only the p_{box} controller in our simulations.

TABLE I
Simulation parameters.

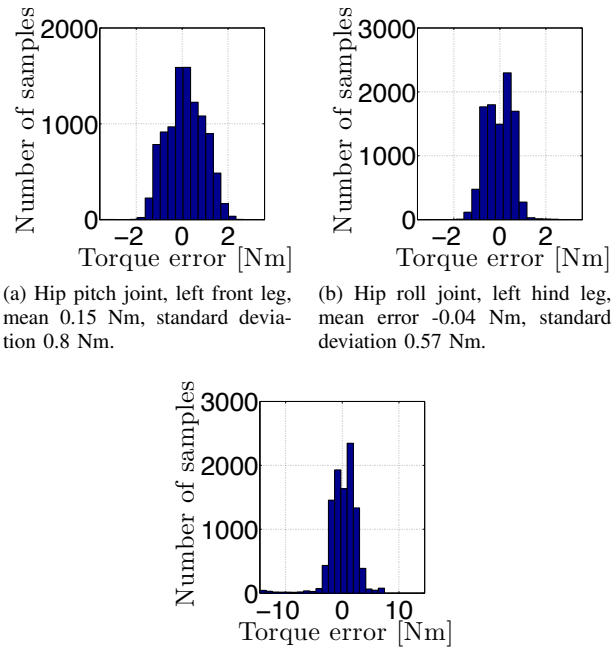
Symbol	Meaning	Value
Δt	Simulation/control time step	2 ms
μ	Force friction coefficient	0.3
v_j^{max}	Max joint velocity	$9.8\,\text{rad}\,\text{s}^{-1}$
$\epsilon_{accuracy}$	Nonlinear solver accuracy	10^{-6}
t_{max}	Max computation time	0.8 ms

The formulation of the control problem changes slightly from test to test, depending on which tasks are implemented. However, if not stated otherwise, the inequality constraints are: linearized friction cones for each contact force, torque limits (upper and lower bounds), and joint acceleration limits (representing the joint-velocity limits). We tuned all the task weights and the feedback gains on the classic controller and then used the same values for the robust controllers. We set all proportional gains k_p between 1 and 100 (e.g. 30 for the CoM, 100 for the swing foot). We set all derivative gains to $k_d = 2\sqrt{k_p}$ to get convergence as fast as possible while avoiding oscillations.

A. Simulation Environment

To assess the proposed controllers we developed a dedicated simulation environment based on a state-of-the-art algorithm for frictional contacts in multibody systems [36]. We integrated the equations of motion of the system with a first-order Euler scheme with fixed time step Δt. Our choice of not using an off-the-shelf simulator is motivated by our desire to completely understand and control the simulation environment. This allowed us to introduce several uncertainties/noises and to regulate their magnitude:

- We added Gaussian noise to the joint torques. The noise had standard deviation σ proportional to the relative maximum torque τ^{max}. To make this noise more realistic (white noise can not exist in the real world) we filtered it using a first-order low-pass filter with a cut frequency of 20 Hz—with a compensation for the amplitude reduction caused by the filter.
- We limited the bandwidth of the torque controller by low-pass filtering the desired joint torques before given them to the simulator. Torque-tracking bandwidths between 40 Hz and 60 Hz have been reported for high-performance actuators (e.g. 40 Hz for hydraulic actuators [17], 46 Hz for electric motors with harmonic drives [37], 60 Hz for series elastic actuators [38]). We assumed a pessimistic torque bandwidth of 20 Hz.
- We estimated joint and base velocities with a Savitsky-Golay filter [39] (fitting a polynomial of order 2 to a sliding window of 20 samples): this introduced a realistic



(a) Hip pitch joint, left front leg, mean 0.15 Nm, standard deviation 0.8 Nm.

(b) Hip roll joint, left hind leg, mean error -0.04 Nm, standard deviation 0.57 Nm.

(c) Knee joint, right hind leg, mean error 0.16 Nm, standard deviation 2.44 Nm.

Fig. 6. Distribution of the joint-torque tracking errors of three joints of the HyQ robot [40], [41].

delay of 20 ms in the velocity signal used by the controller.

- The inertial parameters (masses, centers of mass and inertias) of the model used by the controller did not match those of the model used by the simulator. The random inertial-parameter errors were generated using uniform distribution. For masses and inertias the maximum error was expressed in terms of percentage of the real values. For the centers of mass (CoM) the maximum error was instead expressed in cm.

In each test we specify which uncertainties were simulated.

B. Joint-Torque Tracking Errors

This work is based on the assumption that torque tracking errors are an important uncertainty to take into account in robot controllers. The recent literature on the subject [17], [37], [38] seems to agree with us. To back our assumption we present here some data collected on a real torque-controlled robot. Fig. 6 shows the distribution of the torque tracking errors on three joints of the Hydraulic Quadruped HyQ [40], [41] during a locomotion task. We can clearly see that these errors have approximately zero mean and their distribution can be reasonably approximated by a Gaussian. For this dataset, the standard deviations of the torque tracking errors were between 0.5% and 2% of the maximum joint torque. Given the high performance of the torque controller of HyQ due to the hydraulic actuators and the fast torque feedback we can not hope to achieve the same performance on HRP-2. This is why in our tests we assumed higher values for the standard deviation of the torque errors, between 5% and 10% of the maximum joint torque.

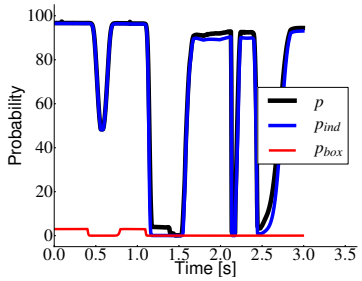


Fig. 7. Test 1. Probability of the inequalities of the TSID control problem computed by the two approximations proposed in this paper.

TABLE II
RESULTS OF TEST 2. FOR EACH FORMULATION WE REPORT THE AVERAGE VALUES OVER 700 TESTS.

Formulation	Classic TSID $w_f = 0$	Classic TSID $w_f > 0$	Robust TSID p_{ind}	Robust TSID p_{box}
Probability $p(\tau)$	25.1	27.4	75.7	66.5
Force prob.	28.2	31.0	86.0	72.2
Joint-accelerat. prob.	85.2	85.2	85.2	85.3
Joint-torque prob.	100	100	100	100
Active inequalities	3.37	2.83	0.6	0.95
Iterations	1.06	1.05	2.06	1.11
Comput. time [ms]	0.23	0.19	0.31	0.2

C. Test 1 — Comparing Probability Approximations

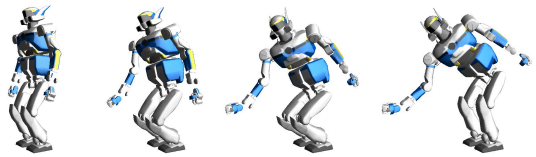
This test aims to compare the different approximations of the probability to satisfy a set of linear inequalities subject to additive noise on the decision variables. We performed this comparison on the TSID inequality constraints, since we are actually interested in how the proposed approximations perform on this particular problem. We generated a state trajectory (i.e. configuration and velocity) by controlling the motion of the CoM of the robot with classic TSID (10). For each state, we computed the probability $p(\tau)$ of the joint torques to satisfy the inequality constraints. We purposely asked for a demanding motion of the CoM (20 cm in 1.6 seconds), which caused several constraints to be saturated, so that $p(\tau)$ covered the whole range $0 - 100$ (see Fig. 7). We then compared $p(\tau)$ with the two approximations p_{ind}, and p_{box}. While p_{ind} is always quite close to p, p_{box} is often far below p. The average error $|p - p_{...}|$ is 2.6% for p_{ind}, and 68% for p_{box}. Despite the poor quality of the p_{box} approximation, the next tests will show that maximizing p_{box} can lead to great improvements in robustness.

D. Test 2 — Comparing Robustness

The goal of this second test is to compare the different TSID formulations in terms of robustness of the inequality constraints. We used the same state trajectory generated for Test 1. For each state we solved the associated control problems using several TSID controllers and we measured the resulting $p(\tau)$ for each. In this test we did not introduce any

TABLE III
RESULTS OF TEST 3. WE REPORT MEAN AND STANDARD DEVIATION OF THE COMPUTATION TIME FOR EACH TESTED TSID FORMULATION.

Formulation	Classic TSID	Robust TSID p_{ind}	Robust TSID p_{box}
Probability $p(\tau)$	26.04	99.23	82.96
Mean comput. time [ms]	0.18	0.33	0.24
Std. dev. comput. time [ms]	0.34	0.42	0.36
Mean comput. time (no warm start) [ms]	10.06	9.36	11.55
Std. dev. comput. time (no warm start) [ms]	0.64	0.95	0.88



(a) 0 s (b) 1 s (c) 2 s (d) 7 s

Fig. 8. Test 3. Snapshots of a reaching task.

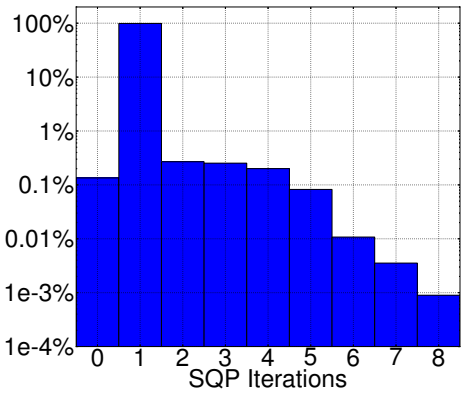


Fig. 9. Test 3. Histogram showing how many times (in percentage) a certain number of (complete) SQP iterations was necessary to converge. One complete iteration consisted in: i) solving a QP to compute the Newton's step, ii) checking convergence, iii) updating the current solution through a line search. The zero-iteration bar corresponds to cases in which the previous control-cycle solution was reused without any modification.

uncertainty/noise in the simulation, but we just measured the probability to satisfy the inequality constraints. The control problem was composed by the following tasks:

- track the desired CoM trajectory (weight 1)
- maintain initial joint posture (weight 10^{-3})
- maximize robustness (weight 10^{-5}, only for robust controllers)
- minimize contact moments and tangential forces [29] (weight w_f, only for classic controller)

The four controllers used in this test are:

- Classic TSID with $w_f = 0$, formulation (10)
- Classic TSID with $w_f = 10^{-4}$, formulation (10)
- Robust TSID with p_{ind} approximation, formulation (14)
- Robust TSID with p_{box} approximation, formulation (19)

Table II reports the results. In terms of probability to satisfy the inequalities, the robust formulations greatly outperform the classic formulations. The force regularization (i.e. $w_f > 0$) slightly improves the overall probability. The optimization of p_{box} leads to a probability slightly lower than p_{ind}, which we expected because of its simplicity. Robust and classic formulations differ the most in the probability of the force inequalities. All the formulations lead to small errors for the CoM task ($< 10^{-3}$).

E. Test 3 — Comparing Computation Times

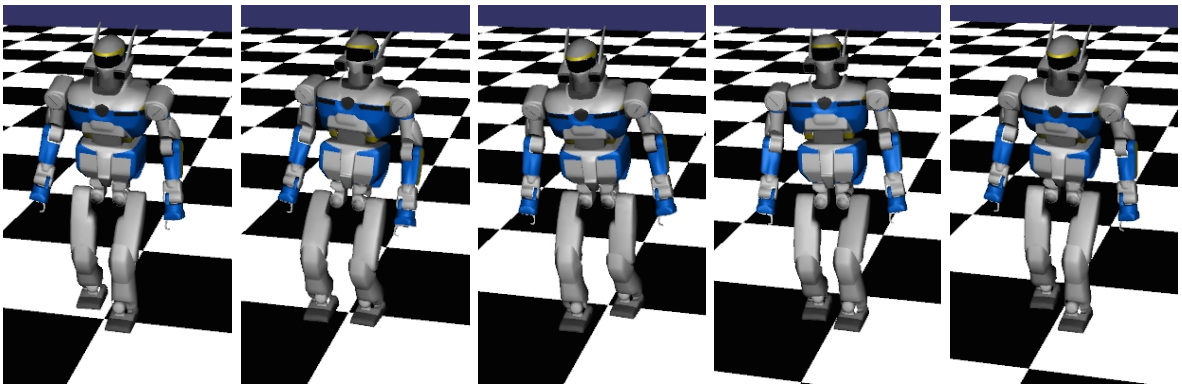
This test focuses on the computation time of the proposed controllers in a whole-body reaching task. In this test we did not introduce any uncertainty/noise in the simulation. The robot had to reach a point far in front with the right hand (see Fig. 8). To avoid falling we constrained the capture point of the robot to lie inside the support polygon [28]. We ran a simulation for each solver, in which we used its solution (i.e. τ) to simulate the system and get its new state.

To speed-up the computation we exploited the warm-start capabilities of *qpOases* [42], the active-set QP solver that we used. To solve the nonlinear problem (14) we implemented a Sequential Quadratic Programming (SQP) algorithm [22]. We initialized the SQP search with the last solution, which most of the times led to convergence in a single Newton's iteration (see Fig. 9). We used a line-search algorithm that enforces strong Wolfe conditions [22]. The algorithm stopped as soon as the squared Newton decrement [35] was less than the desired accuracy ($\Delta x_{newton}^{\top} H \Delta x_{newton} < 2\epsilon_{accuracy}$) or the computation time exceeded an arbitrary limit t_{max}. The computation time only included the time taken by *qpOases*, which means that it neglects the line search and the computation of Hessian and gradient of the cost function. This choice was motivated by two facts. First, the time to solve the QP typically dominates the time taken by the other operations. Second, these operations were implemented in Python, so their computation time is much longer than it would be with a C++ implementation (which will be mandatory for its application on a real robot). Table III shows the results: thanks to the warm start we got an average speed-up of $\sim 30\times$. Apart from a few outliers (maybe due to the Python interface of *qpOases*), the computation time was always below $t_{max} = 0.8$ ms.

F. Test 4 — Walking

To measure robustness in a more concrete way we introduced different uncertainties in our simulation environment and we tested the proposed controllers in it. The target motion consisted in walking on flat ground (see Fig. 10), composed by a forward-walk phase of about 27 seconds, followed by a lateral-walk phase of about 20 seconds. The desired trajectories for the CoM and the feet of the robot were computed using a state-of-the-art walking pattern generator [43]. The control problem was composed by the following tasks:

- track the desired CoM and stepping foot trajectories (weight 1)
- maintain initial joint posture (weight 10^{-3})
- maximize robustness (weight 10^{-4}, only for robust controllers)
- maintain previous joint torques (weight 10^{-5}, only for robust controllers)



(a) 11.6 s (b) 12.0 s (c) 12.4 s (d) 12.8 s (e) 13.2 s

Fig. 10. Test 4. Snapshots of the walking task.

TABLE IV

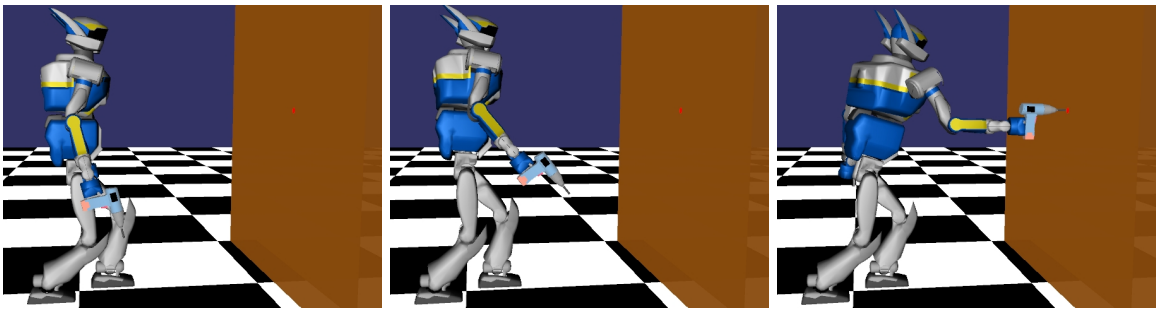
RESULTS OF TEST 4, WALKING. EACH LINE REFERS TO A BATCH OF 100 TESTS UNDER DIFFERENT UNCERTAINTY CONDITIONS. STARTING FROM THE LEFT COLUMN, WE REPORT I) WHETHER THE CONTROLLER USED THE REAL OR THE ESTIMATED VELOCITIES, II) THE STANDARD DEVIATION OF THE GAUSSIAN NOISE ON THE JOINT TORQUES, III) THE TORQUE BANDWIDTH OF THE TORQUE CONTROL, IV) THE MAXIMUM ERROR ON THE MASSES, CENTERS OF MASS AND INERTIAS OF THE LINKS OF THE ROBOT. THE "MEAN TIME BEFORE FALLING" IS THE TOTAL TEST TIME DIVIDED BY THE NUMBER OF FALLS. NOTE THAT THE FIRST FOUR LINES REFER TO SINGLE TESTS BECAUSE THE SIMULATED UNCERTAINTIES ARE DETERMINISTIC.

Uncertainties						Mean time before falling [s]			Number of falls		
v	$\frac{\sigma}{\tau^{max}}$ [%]	Torque bandwidth [Hz]	mass [%]	com [cm]	inertia [%]	Classic	Robust p_{ind}	Robust p_{box}	Classic	Robust p_{ind}	Robust p_{box}
Real	0	∞	0	0	0	∞	∞	∞	0	0	0
Estimated	0	∞	0	0	0	∞	∞	∞	0	0	0
Real	0	20	0	0	0	∞	∞	∞	0	0	0
Estimated	0	20	0	0	0	16.8	∞	20.5	100	0	100
Real	6	∞	0	0	0	203.2	∞	∞	20	0	0
Estimated	6	∞	0	0	0	109.4	912.1	202.5	39	4	20
Estimated	8	∞	0	0	0	69.0	408.3	105.3	58	10	38
Real	6	20	0	0	0	172.6	908.4	∞	23	5	0
Estimated	6	20	0	0	0	24.7	147.3	35.8	98	28	80
Real	0	∞	10	1	20	282.1	∞	∞	15	0	0
Estimated	6	∞	0	0	20	106.1	921.2	240.4	40	4	17
Estimated	6	∞	0	0	100	109.1	761.4	187.2	39	5	22
Estimated	6	∞	10	1	20	94.0	765.7	100.3	44	5	38
Estimated	8	∞	10	1	20	59.0	316.1	102.1	65	14	40
Estimated	5	20	10	1	20	30.8	148.2	33.7	90	28	79

- minimize contact moments and tangential forces (weight 10^{-5}, only for classic controller)

We used a small weight for the robustness to ensure a good tracking of the CoM and the feet, which is critical for the stability of the robot during walking. The task of maintaining the previous joint torques helps having smooth torque trajectories with lower bandwidth than our simulated torque controller (i.e. 20 Hz). We did not include this term in the classic controller because it does not impact its performance. This is due to the minimization of contact moments and tangential forces, which already leads to smooth torque trajectories.

We carried out several batches of tests, each batch differing for the simulated uncertainties. In particular we experimented with the delay in the velocity estimation, the level of noise on the joint torques and the magnitude of the errors in the inertial parameters. We do not report results with different cut frequencies of the torque low-pass filters because they did not seem critical for this test. Each batch was composed by 100 tests, which is not enough for being a statistically significant sampling, but was dictated by the computation time of our simulation environment (about 24 hours for 100 tests). Each test consisted in trying to perform the whole walking motion with three controllers (classic, robust p_{ind}, robust p_{box}) until the robot either fell or reached the end of the motion. We consider that the robot has fallen if the tracking error of its CoM is larger than 50 cm. The random torque noises and the inertial parameter errors changed at each test, but they were the same for the three controllers (i.e. in each test the noise value at every time sample is the same for the three controllers: $e_{classic}(t) = e_{ind}(t) = e_{box}(t)$). We then measured the number of times each controller drove the robot to a fall and the average time before falling (given by the total walking time divided by the number of falls). Table VI-E summarizes the results. When the level of uncertainties is negligible the three controllers perform great, but in the presence of significant uncertainties we see a remarkable difference between the robust controllers and the classic one. It is especially the robust p_{ind} controller that performs much better than the others, reporting much less falls than the others.



(a) 0.0 s (b) 0.6 s (c) 2.6 s

Fig. 11. Test 5. Snapshots of the drilling task. The red dot on the wall represents the desired hole position.

TABLE V

RESULTS OF TEST 5, DRILLING. EACH LINE REFERS TO A BATCH OF 100 TESTS UNDER DIFFERENT UNCERTAINTY CONDITIONS. STARTING FROM THE LEFT COLUMN, WE REPORT I) WHETHER THE CONTROLLER USED THE REAL OR THE ESTIMATED VELOCITIES, II) THE STANDARD DEVIATION OF THE GAUSSIAN NOISE ON THE JOINT TORQUES, III) THE TORQUE BANDWIDTH OF THE TORQUE CONTROL. THE "MEAN TIME BEFORE FALLING" IS THE TOTAL TEST TIME DIVIDED BY THE NUMBER OF FALLS. NOTE THAT THE FIRST TWO LINES REFER TO SINGLE TESTS BECAUSE THE SIMULATED UNCERTAINTIES ARE DETERMINISTIC.

Uncertainties						Mean time before falling [s]			Number of falls		
v	$\frac{\sigma}{\tau^{max}}$ [%]	Torque bandwidth [Hz]	mass [%]	com [cm]	inertia [%]	Classic	Robust p_{ind}	Robust p_{box}	Classic	Robust p_{ind}	Robust p_{box}
Real	0	∞	0	0	0	∞	∞	∞	0	0	0
Estimated	0	∞	0	0	0	∞	∞	∞	0	0	0
Real	5	∞	0	0	0	235.5	1198.9	599.3	5	1	2
Estimated	5	∞	0	0	0	235.4	∞	∞	5	0	0
Estimated	5	20	0	0	0	143.0	590.9	393.8	8	2	3
Estimated	5	20	10	1	20	129.3	297.7	297.8	9	4	4
Real	10	∞	0	0	0	15.7	56.3	36.2	63	20	30
Estimated	10	∞	0	0	0	14.8	56.7	36.6	64	20	30

G. Test 5 — Drilling

This last test is a variation of the previous test, dedicated to a whole-body manipulation task, namely drilling a hole of 5 cm in a wall. Fig. 11 shows some snapshots of the task. While being similar to the whole-body reaching task used in Test 3, this task adds the challenge of physically interacting with the wall. The reaction force exerted by the wall on the drill's tip was simulated with a viscous model, i.e. the force is proportional to the velocity. The control problem was composed of the following tasks:

- maintain the capture point within the support polygon (constraint)
- track the desired 6d trajectory with the drill's tip (weight 1, only when not in contact with the wall)
- track the desired 6d force with the drill's tip (weight 1, only when in contact with the wall)
- maintain initial joint posture (weight 10^{-2})
- maximize robustness (weight 10^{-4}, only for robust controllers)
- maintain previous joint torques (weight 10^{-5}, only for robust controllers)
- minimize contact moments and tangential forces (weight 10^{-6}, only for classic controller)

Once the drill touched the wall we controlled the force in the normal direction (applying a constant force of 50 N that smoothly increases/decreases at the beginning/end), while controlling the drill in impedance in the other 5 directions (i.e. the 3 angular and the 2 tangential linear directions) to avoid drifting. Note that even if in general the capture point is not valid in multi-contact scenarios, given the relatively low values of the contact force exerted on the wall (below 50 N), the capture point remains a reasonable criterion of dynamic equilibrium in this test.

Similarly to the previous test, we carried out several batches of 100 tests each, each batch simulating different uncertainties. Our main objective was to see how often the robot fell during the execution of the task, depending on the controller. Fig. 11 shows that we positioned the robot at a large distance from the wall, so that it had to move its CoM close to the borders of its support polygon to reach the wall with the drill. As expected, the three controllers behaved perfectly as long as the level of the uncertainties was negligible (see first two lines of Table VI-F). However, as soon as we introduced significant noise levels in the simulation the robust controllers outperformed the classic one. Once again, the robust p_{ind} controller performed the best.

H. Discussion

Analyzing the presented results we can claim that:

- optimizing the proposed probability approximations results in a large increase in the real probability to satisfy the inequality constraints;
- the proposed controllers are suitable for online control because the associated optimization problem could be solved in less than 1 ms by exploiting standard warm-start techniques;

- in the presence of Gaussian noise on the joint torques, the robust controllers greatly outperformed the classic controller in walking and whole-body manipulation tasks;
- even in the presence of unmodeled noises and uncertainties (inertial parameters, velocity estimation, limited torque bandwidth), the robust controllers greatly outperformed the classic controller;
- under all uncertainty conditions (except one) the robust p_{ind} controller outperformed the robust p_{box} controller.

It is worth noting that the robust controllers did not outperform the classic controller in every single test, but they always did it (on average) in every batch of 100 tests. This result is reasonable because TSID is a class of instantaneous controllers, meaning that they do not foresee the future behavior of the system. The robust controllers are not able to foresee whether the choice of being robust now will lead the system to a state in which it is no longer possible to be robust. Our tests showed that, on average, choosing to be robust in the present pays off in the future, but we can not guarantee that this is always the case. For instance, during walking, we saw that using a large weight w for the robustness maximization would lead to poor tracking of the center of mass of the robot, which eventually would lead to a fall.

Another interesting result is the superiority of the stochastic robust controller (p_{ind}) with respect to the worst-case robust controller (p_{box}). The stochastic controller knows the probability distribution of the torque noise, which gives this approach a great advantage with respect to the worst-case controller—which knows only its bounds. For instance, in case the robot reaches a state in which one inequality has to be satisfied with zero margin, the worst-case controller must set $s = 0$ and so all solutions would have the same robustness (i.e. zero) for it. This is not the case for the stochastic controller: even if one inequality has zero margin (i.e. it has probability 50%) the stochastic controller would still try to increase the margin of the other inequalities to increase the overall inequality probability. Apparently, this capability to maximize the margin of all the inequalities, rather than to maximize only the smallest margin, results in better performance even in the presence of unmodeled uncertainties. However, the price paid by the stochastic controller is solving a nonlinear optimization problem, which is more complex than the quadratic problem solved by the worst-case solver.

The one case in which the robust p_{box} controller outperforms the p_{ind} controller is probably because of the small number of tests performed. In other words, we believe that by running much more tests under the same noise conditions p_{ind} should outperform p_{box} on average.

1) Common failure modes leading to fall: Looking at the accompanying video one can notice that the falls of the robots look completely different from real-world falls. Of course, this is due to the simulation inaccuracies. We believe that friction (static friction in particular) plays a big role in this regard, but unfortunately accurate friction simulation is extremely challenging and it certainly falls out of the scope of this work. Nonetheless, even if our simulations are not 100% realistic (especially when the robot goes unstable), we can gather some useful information from them. Most of the times, what led to a fall was a contact force violating the friction cone constraints, which in turns led to a foot slipping on the ground. Indeed Table II shows that the probability of the force inequalities is lower than the probability of the acceleration and torque inequalities—and this holds true in all our tests.

Another interesting information is that a perfect velocity feedback seems to significantly help the controllers to recover from foot slippage. For instance, we can see this comparing lines 5 and 6 of Table IV: when subject to the same torque noise, the three controllers (classic, stochastic, worst-case) made the robot fall only $20, 0, 0$ times (respectively) if using a perfect velocity feedback, compared to $39, 4, 20$ times if using a delayed velocity estimation. Thanks to the velocity feedback the controllers can detect foot slippage fast enough to compensate for it and avoid the fall. The same can be said regarding the torque bandwidth (e.g. comparing lines 6 and 9 in Table IV, which differ only for the torque bandwidth). Thanks to the infinite torque bandwidth the controllers are capable of faster reactions when trying to compensate for foot slippage. In conclusion, our simulations suggest that avoiding/compensating for foot slippage is critical to avoid falls. A smart choice could then be to equip the robot's feet with as much sensing capabilities as possible (i.e. force/torque sensors, accelerometers, gyroscopes, tactile sensors) to maximize the quality of the contact force tracking and minimize the reaction time in case of slippage.

VII. Related Works

Considering the robustness of the control scheme is a long-standing and well-identified problem, but it remains largely unanswered. In the literature we can mainly identify three types of approaches:

1) ensuring *stability* despite the presence of undefined uncertainties (typically leveraging control-theory tools);
2) improving robustness to undefined uncertainties through intuitive hand-tunable heuristics;
3) guaranteeing either *stability* or *feasibility* despite the presence of well-identified and modeled uncertainties (typically leveraging robust optimization techniques).

The first approach focuses exclusively on the stability of the system rather than on the feasibility of the state trajectories. For instance, adaptive control [44] and time-delay estimation [45] try to estimate and compensate online for the major errors between nominal and real dynamic model. Other approaches, such as virtual model control [46], do not rely at all on the dynamic model of the robot, which ensures robustness to misestimations of the inertial parameters [47]. Alternatively, robustness can be expressed by considering the stability of the computed trajectories inside a locomotion cycle [48], [49]. The main issue of these schemes is that they do not consider inequality constraints, that is they neglect the problem of feasibility. This makes them hard to implement on real legged robots, given the large number of bounds to which these systems are subject. The sole approach in this category that can handle inequality constraints is the recent work based on sum-of-squares optimization and Lyapunov functions [50]. However, nowadays this approach seems to be applicable only

to systems with a few degrees of freedom, much smaller than humanoid robots.

The second class of works propose hand-tunable heuristics that often are not even emphasized in the papers, but turn out to be fundamental in real implementations. For instance, a common heuristic in TSID—which we adopt as well—is to impose a secondary objective to keep the robot posture close to a reference one [6]. This tends to keep the movement away from the joint limits and therefore it increases robustness. Similarly, to increase the robustness of the contact forces/moments to avoid slipping/tipping, it was proposed to minimize the contact moments and the tangential contact forces in the null space of the main motion task [51]. Yet another common trick in the generation of walking motion is to try to maintain the center of pressure as close as possible to the center of the foot [52]. The robotics literature is filled with these kinds of heuristics, which often are the main reason behind the successful implementations of control algorithms on real platforms. However, these heuristics can not ensure feasibility in the presence of any significant uncertainty and needs ad-hoc tuning depending on the situation.

Finally, the third class of works—which includes this paper—makes use of robust optimization techniques to formulate control and planning problems. Mordatch et al. [53] considered several perturbed models of a humanoid robot to plan offline a trajectory that is robust to uncertainties, reporting success rate between 80% and 95% on a real platform. Another recent work [54] has combined robust and time-scaling optimization to plan trajectories that are robust to bounded errors in friction coefficients and joint accelerations, whose magnitude can be estimated online through iterative learning. Finally, Nguyen and Sreenath [55] have recently exploited control Lyapunov functions and QPs to ensure stability despite bounded uncertainties in the linearized system dynamics. This work [55] is the closest to ours because it deals with online control under bounded uncertainties, and it accounts for inequality constraints (only torque bounds). However, several differences can be noted. First, we focus on the robustness of the inequality constraints, while they focused on stability, which they formulate as a *relaxed* inequality constraint. Second, we proposed to model the uncertainties either as bounded variables or as random variables, while they considered only the former approach. Third, we considered additive errors in the joint-torque tracking, for which identifying a model (either in terms of bounds or probability distribution) seems straightforward. Instead they considered additive errors in the linearized system dynamics, but they gave no details [55] regarding how to find reasonable bounds for this error on a physical system, which does not seem trivial.

VIII. Conclusions

This paper presented an extension of the Task-Space Inverse Dynamics control framework that takes the robustness of the inequality constraints into account. This work is motivated by the low level of robustness that this control framework exhibits in the presence of uncertainties. The proposed solution consists in accounting for additive uncertainties on the joint torques of the robot. Our choice to focus on the joint-torque errors is motivated by the challenge of having good joint-torque tracking [18] on real robots and by the tractability of the resulting mathematical problem. We showed that these uncertainties can be either modeled as random variables with known probability distribution—giving rise to a stochastic optimization problem— or as deterministic bounded variables—resulting in a worst-case optimization problem. Since the resulting stochastic optimization problem is too complex to solve in few milliseconds we proposed two ways to approximate this problem, one of which turned out to be almost identical to the case of deterministic uncertainties. Through extensive simulations in a *realistic* environment we tested the proposed robust controllers against the classic (nonrobust) controller. Regardless of the nature and the magnitude of the simulated uncertainties the robust controllers greatly outperformed the classic controller in terms of number of falls of the robot.

Of course real robots are affected by many more uncertainties than those that we introduced in our environment, such as errors and delays in the estimation of the floating-base position-orientation, static friction at the joints, communication delays, errors in the geometric parameters, link flexibility. While we did not account for all these uncertainties, the uncertainties that we introduced in our simulations were sufficient to show the potential benefits of our approach. Moreover, it is reasonable to assume that the robust controllers would always outperform (on average) the classic controller, regardless of the source of the uncertainties.

A. Future work

It would be easy to generalize the presented results to the case of nonzero-mean and nondiagonal covariance matrix; we did not deal with these cases because we found these two assumptions very reasonable in practice. We could also use a different probability distribution as long as we can compute the probability density function and cumulative density function of a weighted sum of random variables following such a distribution. Unfortunately, this operation is trivial only for the Gaussian and a few other simple distributions [19], [56], [57].

Another straightforward extension of this work would be to immunize to uncertainties the stability of the system—besides the feasibility of the state-control trajectories. Using control Lyapunov function [55] we could represent tasks as inequalities—rather than equalities—and apply the same methodology discussed in this paper.

A more challenging extension would be to consider other types of uncertainties. The proposed controllers immunize the inequality constraints only to joint-torque tracking errors. Even if they also exhibited an increased level of robustness to other unmodeled uncertainties, taking into account all of the uncertainties affecting the system would result in better performance. Some other uncertainties that appear linearly in the constraints exist, for instance additive uncertainties in the joint velocities appear linearly in the capture-point constraints. However, many uncertainties appear nonlinearly in the dynamics of the system (e.g. in the forward dynamics

joint velocities appear quadratically, and inertial parameters multiply joint accelerations), which prevents us from dealing with them using the same optimization techniques discussed in this paper. However, we can hope to do that leveraging nonlinear robust optimization techniques [58], [59].

We believe that accounting for uncertainties could lead to major improvements in robotics if applied at all levels (i.e. planning, control, estimation and identification). For instance, the same approaches presented here could be applied to Model Predictive Control (MPC), a control technique that has become ubiquitous in robotics for the generation of walking motion [43], [52]. While robust MPC is already an active research field [60]–[62], applications of robust optimization in robotics are seldom [54], [55], [63], [64].

While this work focused on the theoretical results and their validation, its motivations lie in the desire to control real robots. We plan to test the presented control algorithms on HRP-2 (on which we recently implemented torque control [18]) and empirically measure the robustness improvements. We are also interested in testing robust TSID with strict priorities, which will require the implementation of a hierarchical nonlinear convex solver [27].

ACKNOWLEDGMENT

This work was supported by FP7 EU projects Euroc (Grant Agreement 608849), Koroibot (Grant Agreement 611909) and Entracte (ANR Grant Agreement 13-CORD-002-01). The authors would like to thank Michele Focchi, Steve Tonneau, Florent Lamiraux, and Maximilien Naveau for the help and the useful discussions.

REFERENCES

[1] T. Koolen, J. Smith, G. Thomas, S. Bertrand, J. Carff, N. Mertins, D. Stephen, P. Abeles, J. Englsberger, S. Mccrory, J. V. Egmond, M. Griffioen, M. Floyd, S. Kobus, N. Manor, S. Alsheikh, D. Duran, L. Bunch, E. Morphis, L. Colasanto, K.-l. H. Hoang, B. Layton, P. Neuhaus, M. Johnson, and J. Pratt, "Summary of Team IHMC ' s Virtual Robotics Challenge Entry," in *Humanoid Robots (Humanoids)*, 2013.

[2] S. Faraji, S. Pouya, C. G. Atkeson, and A. J. Ijspeert, "Versatile and Robust 3D Walking with a Simulated Humanoid Robot (Atlas): a Model Predictive Control Approach," in *Robotics and Automation (ICRA), IEEE International Conference on*, 2014, pp. 1943–1950.

[3] A. Herzog, L. Righetti, F. Grimminger, P. Pastor, and S. Schaal, "Balancing experiments on a torque-controlled humanoid with hierarchical inverse dynamics," in *Intelligent Robots and Systems (IROS), 2014 IEEE/RSJ International Conference on*, no. Iros, 2014.

[4] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, "Operational Space Control: A Theoretical and Empirical Comparison," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 737–757, jun 2008.

[5] F. Nori, S. Traversaro, J. Eljaik, F. Romano, A. Del Prete, and D. Pucci, "iCub Whole-Body Control through Force Regulation on Rigid Non-Coplanar Contacts," *Frontiers in Robotics and AI*, vol. 2, 2015.

[6] A. Del Prete, F. Nori, G. Metta, and L. Natale, "Prioritized Motion-Force Control of Constrained Fully-Actuated Robots: "Task Space Inverse Dynamics"," *Robotics and Autonomous Systems*, 2014.

[7] L. Saab, O. E. Ramos, N. Mansard, P. Soueres, and J.-y. Fourquet, "Dynamic Whole-Body Motion Generation under Rigid Contacts and other Unilateral Constraints," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 346–362, 2013.

[8] L. Sentis and O. Khatib, "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives," *International Journal of Humanoid Robotics*, vol. 2, no. 4, pp. 505–518, 2005.

[9] O. E. Ramos, N. Mansard, O. Stasse, J.-b. Hayet, and P. Soueres, "Towards reactive vision-guided walking on rough terrain: an inverse-dynamics based approach," *International Journal of Humanoid Robotics*, vol. 2, 2014.

[10] Y. Abe, M. da Silva, and J. Popović, "Multiobjective control with frictional contacts," *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 249–258, 2007.

[11] M. De Lasa, I. Mordatch, and A. Hertzmann, "Feature-based locomotion controllers," *ACM Transactions on Graphics*, vol. 29, no. 4, p. 1, jul 2010.

[12] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *International Journal of Robotics Research*, 2014.

[13] G. Pratt, "http://www.theroboticschallenge.org/," 2015.

[14] S. Feng, X. Xinjilefu, C. G. Atkeson, and J. Kim, "Optimization Based Controller Design and Implementation for the Atlas Robot in the DARPA Robotics Challenge Finals," in *Humanoid Robots, 2015 IEEE-RAS International Conference on*, 2015.

[15] M. Johnson, B. Shrewsbury, S. Bertrand, T. Wu, D. Duran, M. Floyd, P. Abeles, D. Stephen, N. Mertins, A. Lesman, J. Carff, W. Rifenburgh, P. Kaveti, W. Straatman, J. Smith, M. Griffioen, B. Layton, T. de Boer, T. Koolen, P. Neuhaus, and J. Pratt, "Team IHMC's Lessons Learned from the DARPA Robotics Challenge Trials," *International Journal of Field Robotics*, vol. 32, pp. 192–208, 2015.

[16] C. Liu, C. G. Atkeson, S. Feng, and X. Xinjilefu, "Full-body Motion Planning and Control for The Car Egress Task of the DARPA Robotics Challenge," Tech. Rep., 2015.

[17] T. Boaventura, C. Semini, J. Buchli, M. Frigerio, M. Focchi, and D. G. Caldwell, "Dynamic torque control of a hydraulic quadruped robot," *2012 IEEE International Conference on Robotics and Automation*, pp. 1889–1894, may 2012.

[18] A. Del Prete, N. Mansard, O. E. Ramos, O. Stasse, and F. Nori, "Implementing Torque Control with High-Ratio Gear Boxes and without Joint-Torque Sensors," *International Journal of Humanoid Robotics (accepted for publication)*, 2015.

[19] R. Henrion, "Introduction to Chance Constrained Programming," *Tutorial paper for the Stochastic Programming Community*, 2004.

[20] A. Ben-Tal and A. Nemirovski, "Robust convex optimization," *Mathematics of Operations Research*, vol. 23, no. 4, pp. 769–805, 1998.

[21] A. Del Prete and N. Mansard, "Addressing Constraint Robustness to Torque Errors in Task-Space Inverse Dynamics," in *Robotics, Science and Systems (RSS)*, Rome, 2015.

[22] S. J. Wright and J. Nocedal, *Numerical optimization*, 1999.

[23] E. A. Yildirim and S. J. Wright, "Warm-Start Strategies in Interior-Point Methods for Linear Programming," *SIAM Journal on Optimization*, vol. 12, no. 3, pp. 782–810, jan 2002.

[24] D. Goldfarb and a. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Mathematical Programming*, vol. 27, no. 1, pp. 1–33, 1983.

[25] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and Applications of Robust Optimization," *SIAM*, p. 50, oct 2011.

[26] D. Bertsimas and M. Sim, "The price of robustness," *Operations research*, 2004.

[27] A. Del Prete, F. Romano, L. Natale, G. Metta, G. Sandini, and F. Nori, "Prioritized Optimal Control," in *Robotics and Automation (ICRA), IEEE International Conference on*, 2014.

[28] O. E. Ramos, N. Mansard, and P. Soueres, "Whole-body Motion Integrating the Capture Point in the Operational Space Inverse Dynamics Control," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2014.

[29] L. Righetti, J. Buchli, M. Mistry, M. Kalakrishnan, and S. Schaal, "Optimal distribution of contact forces with inverse dynamics control," *The International Journal of Robotics Research*, no. January, jan 2013.

[30] A. Genz, "Numerical computation of multivariate normal probabilities," *Journal of computational and graphical statistics*, vol. 1, no. 2, pp. 140–149, 1992.

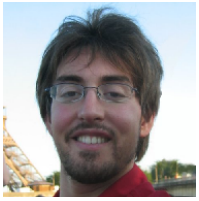
[31] L. Saab, O. Ramos, N. Mansard, P. Souères, and J.-Y. Fourquet, "Dynamic Whole-Body Motion Generation Under Rigid Contacts and Other Unilateral Constraints," *IEEE Trans. on Robotics*, vol. 29, no. 2, apr 2013.

[32] S. Rubrecht, V. Padois, P. Bidaud, and M. De Broissia, "Constraints Compliant Control : constraints compatibility and the displaced configuration approach," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010.

[33] A. Del Prete, N. Mansard, F. Nori, G. Metta, and L. Natale, "Partial Force Control of Constrained Floating-Base Robots," in *Intelligent Robots and Systems (IROS), IEEE International Conference on*, 2014.
[34] H. Vazquez-Leal, "High accurate simple approximation of normal distribution integral," *Mathematical problems in engineering*, 2012.
[35] S. Boyd and L. Vandenberghe, *Convex Optimization*, 2004, vol. 98.
[36] D. M. Kaufman, S. Sueda, D. L. James, and D. K. Pai, "Staggered projections for frictional contact in multibody systems," *ACM Transactions on Graphics*, vol. 27, no. 5, p. 1, 2008.
[37] H. Dallali, G. A. Medrano-cerda, M. Focchi, T. Boaventura, M. Frigerio, C. Semini, J. Buchli, and D. G. Caldwell, "On Use of Positive Feedback for Improved Torque Control," Tech. Rep., 2014.
[38] N. Paine, S. Oh, and L. Sentis, "Design and Control Considerations for High-Performance Series Elastic Actuators," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 3, pp. 1080–1091, 2014.
[39] R. W. Schafer, "What is a Savitzky-Golay filter?" *Signal Processing Magazine, IEEE*, vol. 28, no. 4, pp. 111–117, 2011.
[40] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. G. Caldwell, "Design of HyQ - a hydraulically and electrically actuated quadruped robot," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 225, no. 6, pp. 831–849, 2011.
[41] C. Semini, V. Barasuol, T. Boaventura, M. Frigerio, M. Focchi, D. G. Caldwell, and J. Buchli, "Towards versatile legged robots through active impedance control," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 1003–1020, 2015.
[42] H. J. Ferreau, C. Kirches, and A. Potschka, "qpOASES: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, 2013.
[43] M. Naveau, J. Carpentier, S. Barthelemy, O. Stasse, and P. Soueres, "METAPOD - Template META-PrOgramming applied to Dynamics: CoP-CoM trajectories filtering." in *IEEE/RAS International Conference on Humanoid Robot (ICHR)*, 2014.
[44] R. Kelly and R. Carelli, "On adaptive impedance control of robot manipulators," *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, 1989.
[45] M. Jin, S. H. Kang, and P. H. Chang, "Robust compliant motion control of robot with nonlinear friction using time-delay estimation," *Industrial Electronics, IEEE Transactions on*, vol. 55, no. 1, 2008.
[46] J. Pratt, C.-m. Chew, A. Torres, P. Dilworth, and G. Pratt, "Virtual model control: An intuitive approach for bipedal locomotion," *The International Journal of Robotics Research*, vol. 20, no. 2, pp. 129–143, 2001.
[47] A. Dietrich, C. Ott, and A. Albu-Schaffer, "Multi-objective compliance control of redundant manipulators: Hierarchy, control, and stability," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, nov 2013, pp. 3043–3050.
[48] K. Mombaur, R. Longman, H. G. Bock, and J. Schlöder, "Open-loop stable running," *Robotica*, vol. 23, no. 1, pp. 21–33, jan 2005.
[49] E. R. Westervelt, J. W. Grizzle, and C. Chevallereau, *Feedback control of dynamic bipedal robot locomotion*, 2007.
[50] A. Majumdar, A. A. Ahmadi, and R. Tedrake, "Control Design Along Trajectories via Sum of Squares Optimization," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013.
[51] L. Righetti, J. Buchli, M. Mistry, and S. Schaal, "Inverse Dynamics With Optimal Distribution of Ground Reaction Forces for Legged Robots," in *Emerging Trends in Mobile Robotics - Proceedings of the 13th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*. Singapore: World Scientific Publishing Co. Pte. Ltd., 2010, pp. 580–587.
[52] S. Kajita and F. Kanehiro, "Biped walking pattern generation by using preview control of zero-moment point," in *Robotics and Automation (ICRA), 2003 IEEE International Conference on*, 2003.
[53] I. Mordatch, K. Lowrey, and E. Todorov, "Ensemble-CIO : Full-Body Dynamic Motion Planning that Transfers to Physical Humanoids," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.
[54] J. Luo and K. Hauser, "Robust Trajectory Optimization Under Frictional Contact with Iterative Learning," in *Robotics, Science and Systems (RSS)*, 2015.
[55] Q. Nguyen and K. Sreenath, "Optimal Robust Control for Bipedal Robots through Control Lyapunov Function based Quadratic Programs," in *Robotics, Science and Systems (RSS)*, 2015.
[56] R. Henrion, "Gradient estimates for Gaussian distribution functions: application to probabilistically constrained optimization problems," *Numerical Algebra, Control and Optimization*, vol. 2, no. 4, pp. 655–668, nov 2012.
[57] R. Henrion and A. Möller, "A Gradient Formula for Linear Chance Constraints Under Gaussian Distribution," *Mathematics of Operations Research*, vol. 37, no. 3, pp. 475–488, aug 2012.
[58] Y. Zhang, "General robust-optimization formulation for nonlinear programming," *Journal of Optimization Theory and Applications*, vol. 132, pp. 111–124, 2007.
[59] H.-g. Beyer and B. Sendhoff, "Robust Optimization A Comprehensive Survey," *Computer methods in applied mechanics and engineering*, vol. 196, no. 33, pp. 3190–3218, 2007.
[60] A. Bemporad and M. Morari, "Robust Model Predictive Control : A Survey," *Robustness in identification and control*, pp. 207–226, 1999.
[61] P. Li, M. Wendt, and G. Wozny, "Robust model predictive control under chance constraints," *Computers & Chemical Engineering*, vol. 24, pp. 829–834, 2000.
[62] J. Björnberg and M. Diehl, "Approximate robust dynamic programming and robustly stable MPC," *Automatica*, 2006.
[63] A. Majumdar and R. Tedrake, "Robust online motion planning with regions of finite time invariance," *Algorithmic Foundations of Robotics X*, pp. 1–16, 2013.
[64] C. Brasseur, A. Sherikov, C. Collette, D. Dimitrov, and P.-b. Wieber, "A Robust Linear MPC Approach to Online Generation of 3D Biped Walking Motion," in *Humanoid Robots (Humanoids), 2015 15th IEEE-RAS International Conference on*, 2015.



Andrea Del Prete Andrea Del Prete was born in Cesena (Italy) in 1984. He received his degree in Computer Engineering (with honors) from the 2nd faculty of the University of Bologna (Italy) in 2009. In March 2013 he got his PhD from the Cognitive Humanoids laboratory of the department of "Robotics Brain and Cognitive Sciences" in IIT, Genova. Since 2014 he has been a PostDoc at LAAS-CNRS in Toulouse, working on inverse-dynamics control with HRP-2.



Nicolas Mansard Nicolas Mansard received the joint M.Eng. and M.Sc. degrees in robotics and image processing from the Nationale Supérieure d'Informatique et de Mathématiques Appliquées de Grenoble, Grenoble, France, and the University Joseph Fourier, Grenoble, and the Ph.D. degree for his work with the Lagadic Group, Institut National de Recherche en Informatique et en Automatique, Rennes, France, in 2006. He spent one year with Stanford University, Stanford, CA, with O. Khatib and one year with the JRL-Japan with A. Kheddar. He is currently with the GEPETTO Group, LAAS-CNRS. He was invited research in Emo Todorov's Group at University of Washington in 2013. His research is concerned with generation of dynamic sensor-based robot movements. He received the CNRS Bronze medal in 2015.