



**HAL**  
open science

# Addressing Constraint Robustness to Torque Errors in Task-Space Inverse Dynamics

Andrea del Prete, Nicolas Mansard

► **To cite this version:**

Andrea del Prete, Nicolas Mansard. Addressing Constraint Robustness to Torque Errors in Task-Space Inverse Dynamics. Robotics, Sciences and Systems 2015, Jul 2015, Rome, Italy. hal-01109375v2

**HAL Id: hal-01109375**

**<https://hal.science/hal-01109375v2>**

Submitted on 21 May 2015

**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

# Addressing Constraint Robustness to Torque Errors in Task-Space Inverse Dynamics

Andrea Del Prete, Nicolas Mansard  
 CNRS, LAAS, 7 avenue du colonel Roche,  
 Univ de Toulouse, LAAS, F-31400 Toulouse, France.  
 Email: adelpret@laas.fr, nmansard@laas.fr

**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 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 as additive white random noise (similarly to LQG). This results in a stochastic optimization problem, in which we can maximize the probability to satisfy the inequality constraints (i.e. to be feasible). Since computing this probability is computationally expensive, we propose three ways to approximate it that are much faster to compute and that we can then use for online control (resolution time below 1 ms). Simulation results show that taking robustness into account greatly increases the chances to have feasible control trajectories (even when the uncertainties affecting the system are not the one modeled in the controller).

## I. INTRODUCTION

Task-space inverse dynamics (TSID) has become an increasingly popular way to control humanoid and quadruped robots [1, 2, 3, 4, 5]. This technique offers a number of attractive features: it is theoretically sound [6], it works well in simulation [7, 8], and it is fast enough to be applied for online control. 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.

To improve the robustness of the control trajectories, we propose to account for additive random noise on the joint torques. First, Section II introduces the problem arising from solving an optimization problem without accounting for uncertainties. Then Section II-A focuses on the case in which the problem variable is affected by additive random noise. Unfortunately, the resulting stochastic optimization problem<sup>1</sup> is too computationally expensive to be used for online control. In Sections II-B and II-C, we discuss three different ways to approximate a general stochastic optimization problem with linear inequality constraints. Section III shows how these ideas relate to the specific TSID control problem. We then

<sup>1</sup>In a stochastic optimization problem some of the variables are random and follow a known probability distribution

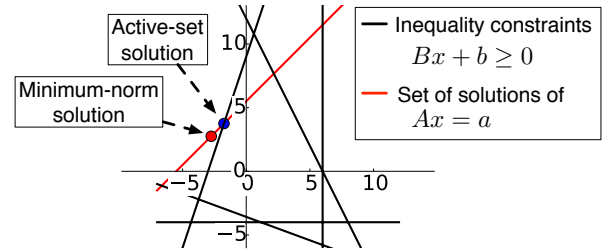


Fig. 1: 2D example of an inequality-constrained least-square problem, such as (1), solved by an active-set method.

discuss how to use the three presented approximations to get different formulations of *robust TSID* (Sections IV and V). Finally, in Section VII we validate the proposed methods on a simulated HRP-2 humanoid robot. We show that taking robustness into account greatly increases the chances to have feasible trajectories for a system subject to uncertainties (even when these uncertainties are not the one modeled in the controller). 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.

## II. FROM DETERMINISTIC TO STOCHASTIC OPTIMIZATION

Let us consider a general constrained least-square optimization problem of the form:

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

In the next section we will see how this relates to the problem of controlling a robot with TSID. In robotics typically (1) has infinite solutions (because  $A$  is a fat matrix) so the determined solution depends on the technique we use to solve it. The most common approach to solve problem (1) in robotics is through active-set methods [9], mainly because they are easy to warm-start<sup>2</sup> (contrary to interior-point methods). Take for instance the 2D example depicted in Fig. 1. The active-set algorithm first computes a solution of the unconstrained problem (i.e. red dot). Since this solution violates an inequality

<sup>2</sup>Warm-starting the resolution of an optimization problem consists in exploiting the solution of a similar problem (which was already computed) to speed-up the computation.

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. Because of their working principle, active-set algorithms tend to find the worst solutions in terms of robustness. 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.

### A. Introducing Random Variables

Our idea is to model uncertainties as Gaussian noise  $e \sim \mathcal{N}(0, \Sigma)$  (with a diagonal covariance matrix<sup>3</sup>  $\Sigma = \text{diag}([\sigma_1^2 \ \dots \ \sigma_n^2])$ ) affecting the problem's variable  $x$ :

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

Since  $e$  is a random variable, both cost and constraints are now random variables, so (2) 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:

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

The inequalities are less trivial and consequently less frequently considered. A classic approach is to replace them with their probability to be satisfied [10]:

$$p(x) = \mathbf{P}(B(x + e) + b \geq 0) \quad (3)$$

Rather than directly using  $p(\cdot)$  in our optimization problem, we define another function,  $R(\cdot)$ , which is easier to optimize than  $p(\cdot)$ <sup>4</sup>. Once we have defined  $R(\cdot)$ , we can use it in three different ways.

1) *Priority to cost*: We can maximize robustness in the null space of the cost:

$$\begin{aligned} & \underset{x \in S}{\text{minimize}} \quad R(x) \\ & S = \{x | Ax = Ax^*\}, \end{aligned} \quad (4)$$

where  $x^*$  is a solution of the non-robust problem (1). This approach does not affect the cost, it only exploits its redundancy to maximize the robustness of the inequality constraints. This means that the solution may have little robustness if that is necessary to get a small cost.

<sup>3</sup>All results can be generalized to any probability distribution, with arbitrary mean and nondiagonal covariance matrices, as long as we can compute its probability density function (pdf) and cumulative density function (cdf).

<sup>4</sup>For instance, rather than maximizing a probability we maximize its logarithm because it is a concave function.

2) *Priority to robustness*: We can minimize the cost subject to the constraint of ensuring a minimum robustness  $R_{min}$ :

$$\begin{aligned} & \underset{x}{\text{minimize}} \quad \|Ax - a\|^2 \\ & \text{subject to} \quad R(x) \leq R_{min} \end{aligned} \quad (5)$$

Then of course we could still maximize robustness in the null space of this minimization. To select  $R_{min}$  so that the constraint is feasible we might first minimize  $R(x)$  to find its minimum value.

3) *Trade off cost and robustness*:

$$\begin{aligned} & \underset{x}{\text{minimize}} \quad \|Ax - a\|^2 + wR(x) \\ & \text{subject to} \quad Bx + b \geq 0, \end{aligned} \quad (6)$$

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).

Regardless of which of these three formulations we use, we need to evaluate the cumulative density function (cdf) of the multivariate random variable  $e_B = Be$ , that is  $\mathbf{P}(e_B > -b - Bx)$ . In general there is no analytical expression to compute this cdf, so we must resort to computationally-expensive numerical techniques [11]. This makes the resolution too slow for applications in control. Aside from introducing robustness in TSID, the main contribution of this paper is to propose three approximations of (3) that are much faster to compute and that provide satisfying precision and robustness in practice.

### B. Approximation 1 — Individual Constraints $p_{ind}$

The first way to simplify (3) is by considering the probabilities of the single inequalities rather than the probability of all of them:

$$p_{ind}(x) = \prod_{i=1}^m \mathbf{P}(B_i(x + e) + b_i \geq 0), \quad (7)$$

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  $e_B$ . This approximation fastens to resolution of (3) 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. Fig. 2a depicts the probability  $p(x)$  to satisfy a set of 5 linear stochastic inequalities, with  $e$  having a standard deviation  $\sigma_1 = \sigma_2 = 1.44$ . Fig. 2b shows the approximated probability  $p_{ind}(x)$  obtained with (7), while Fig. 2d 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. 2c 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  $< 90^\circ$  the error is negative, when the angle is  $> 90^\circ$  the error is positive, whereas when the angle is exactly  $90^\circ$  the error is void.

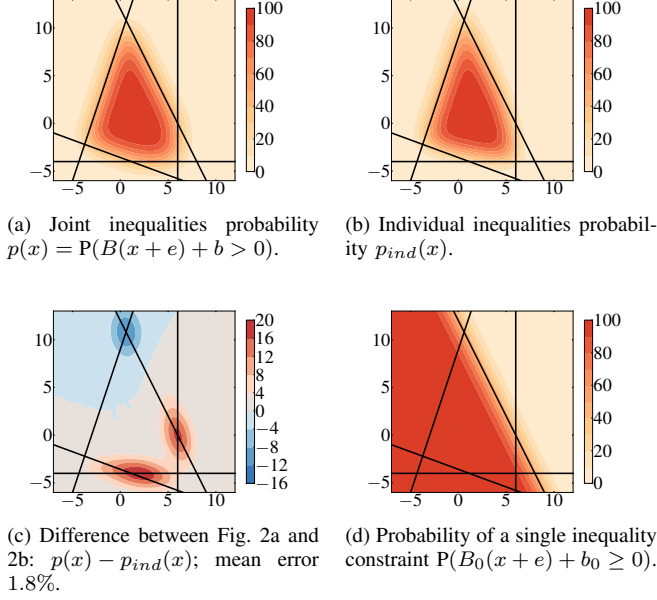


Fig. 2: 2D example comparing  $p(x)$ , namely the probability of satisfying a set of affine stochastic inequalities, with its approximation  $p_{ind}(x)$  (7).

### C. Approximation 2/3 — Largest-Enclosed Hyper-Rectangle $p_{rect}/p_{box}$

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 (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:

$$p(x) \approx \mathbb{P}((x+e) \in U(s)) \quad (8)$$

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  $\mathbb{P}((x+e) \in U(s))$ :

$$\begin{aligned} p_{rect}(x) = \underset{s}{\text{maximize}} \quad & \mathbb{P}((x+e) \in U(s)) \\ \text{subject to} \quad & Bz + b \geq 0 \quad \forall z \in U(s), \end{aligned} \quad (9)$$

where the (infinite) constraints ensure that  $U(s)$  is enclosed in the feasible set. Fig. 3 shows the hyper-rectangle maximizing the probability for different parametrizations  $s$ . In Fig. 3a  $U(s)$  is a general hyper-rectangle and hence, we need  $2n$  values to represent it. In Fig. 3b  $U(s)$  is a hyper-cube so we can represent it with only 1 scalar value (in this case we refer to the probability approximation as  $p_{box}$ ). Clearly these different

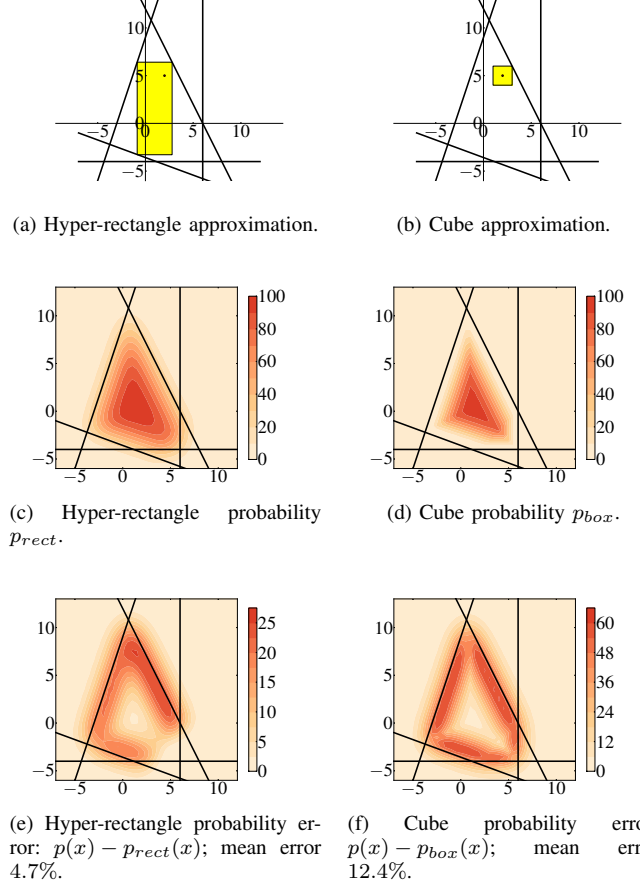


Fig. 3: 2D example comparing  $p(x)$  with its approximations  $p_{rect}(x)$  and  $p_{box}(x)$  (9).

parametrizations trade off the quality of the approximation for its simplicity. Fig. 3c and 3d show the value of  $p_{rect}$  and  $p_{box}$  over the solution space, whereas Fig. 3e and 3f show the approximation errors  $(p(x) - p_{rect}(x))$  and  $(p(x) - p_{box}(x))$ . While these approximations may seem much coarser than the first one, in Section VII we will show empirically that they perform well in practice. Moreover, in Section V we will prove that the single-variable parametrization 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 [6]. We use here the formulation of [12], written as an optimization problem of  $y^\top = [\dot{v}^\top \ f^\top \ \tau^\top]$ :

$$\begin{aligned} \underset{y}{\text{minimize}} \quad & \|Ay - a\|^2 \\ \text{subject to} \quad & By + b \geq 0 \end{aligned} \quad (10)$$

$$\begin{bmatrix} J_c & 0 & 0 \\ M & -J_c^\top & -S^\top \end{bmatrix} \begin{bmatrix} \dot{v} \\ f \\ \tau \end{bmatrix} = \begin{bmatrix} -J_c v \\ -h \end{bmatrix},$$

where  $\dot{v}$  are the base and joint accelerations,  $f$  are the contact forces,  $\tau$  are the joint torques,  $J_c$  is the constraint Jacobian,  $M$  is the mass matrix,  $h$  contains the bias forces and  $S$  is the selection matrix. The inequality constraints (defined by  $B$  and  $b$ ) contain the torque limits, the (linearized) 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 [13]. The cost function represents the error of the task, which is typically an affine function of  $\dot{v}$  (i.e. a task-space acceleration):

$$Ay - a = \underbrace{\begin{bmatrix} J & 0 & 0 \end{bmatrix}}_A y - \underbrace{(\ddot{x}^{des} - \dot{J}v)}_a = \ddot{x} - \ddot{x}^{des}$$

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

This problem is rather similar to the one we considered in the previous section, apart from the fact that it has equality constraints. The uncertainty  $e \in \mathbb{R}^n$  in the joint torques prevents us from selecting a value of  $y$  that satisfies the equality constraints. For this reason, we need to reformulate (10) with respect to  $\tau$  alone by expressing  $\dot{v}$  and  $f$  as functions of  $\tau$ <sup>5</sup>:

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

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

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

where  $D = AC$ ,  $d = a - Ac$ ,  $G = BC$ ,  $g = Bc + b$ . Introducing stochasticity we measure the robustness of a solution of (11) by some approximations of:

$$p(\tau) = \text{P}(G(\tau + e) + g \geq 0)$$

Among the three discussed options to introduce robustness in the optimization (i.e. (4), (5) and (6)) we decided to look for a trade-off between performances and robustness, which is:

$$\begin{aligned} & \underset{\tau}{\text{minimize}} \quad \|D\tau - d\|^2 + wR(\tau) \\ & \text{subject to} \quad G\tau + g \geq 0, \end{aligned} \quad (12)$$

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

#### IV. APPROXIMATION 1 — INDIVIDUAL CONSTRAINTS $p_{ind}$

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

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

<sup>5</sup>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 [14].

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 [15] (e.g. polynomials). To compute  $p_{ind}$  we need to evaluate  $\text{P}(G_i(\tau + e) + g_i \geq 0)$ . Since  $e$  is Gaussian, we have  $e_{G_i} = G_i e \sim \mathcal{N}(0, \sigma_{G_i})$ , where  $\sigma_{G_i} = \sigma_i^2 G_i G_i^T$ . Hence:

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

where  $F_{G_i}(\cdot)$  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 [16] (see the Appendix for the expressions of the gradient and the Hessian of  $R_{ind}$ ). The final robust TSID problem is then a convex optimization:

$$\underset{\tau}{\text{minimize}} \quad \|D\tau - d\|^2 - w \sum_{i=1}^m \log F_{G_i}(G_i\tau + g_i) \quad (13)$$

subject to  $G\tau + g \geq 0$

#### V. APPROXIMATION 2 — LARGEST-ENCLOSED HYPER-RECTANGLE $p_{rect}/p_{box}$

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:

$$\begin{aligned} p_{rect}(\tau) &= \underset{s}{\text{maximize}} \quad \text{P}(e \in U(s)) \\ & \text{subject to} \quad G(\tau + z) + g \geq 0, \quad \forall z \in U(s) \end{aligned} \quad (14)$$

First we will define  $U(s)$  for a general hyper-rectangle, and then we will show how things simplify for the hyper-cube. When  $U$  is a general hyper-rectangle (aligned with the main axes) and consequently  $s \in \mathbb{R}^{2n}$ , we define  $U(s)$  as:

$$U(s) = \{z \in \mathbb{R}^n : s_{i+n} \leq z_i \leq s_i \quad i = 1 \dots n\}$$

According to this definition, the first  $n$  values of  $s$  are the upper bounds, while the second  $n$  values are the lower bounds of the hyper-rectangle. The fact that  $U(s)$  is aligned with the main axes and that  $\Sigma$  is diagonal results in this simplification:

$$\begin{aligned} \text{P}(e \in U(s)) &= \prod_{i=1}^m \text{P}(s_{i+n} \leq e_i \leq s_i) = \\ &= \sum_{i=1}^m (F_i(s_i) - F_i(s_{i+n})), \end{aligned}$$

where  $F_i$  is the cdf of  $e_i$ . Despite this simple way of computing the probability of the inequalities, using  $p_{rect}(x)$  in place of  $p(x)$  may seem rather complex. First, we need to optimize a function ( $p_{rect}$ ) that is itself the solution of an optimization problem and second, (14) cannot be solved in this form since it has an infinite number of constraints. Despite appearances,

this is not more complex to solve than our first approximation; on the contrary, for a particular choice of the parametrization of  $U$ , this boils down to a simple linear program. Let us now show how we can achieve this.

### A. Reduction of the Infinite Number of Constraints

We can represent the infinite constraints of (14) as a finite number of constraints:

$$l_i(s) \geq 0 \quad i = 1 \dots m,$$

where  $l_i$  is the solution of an optimization problem:

$$\begin{aligned} l_i(s) = \underset{z}{\text{minimize}} \quad & G_i(\tau + z) + g_i \\ \text{subject to} \quad & s_{i+n} \leq z_i \leq s_i \quad i = 1 \dots n \end{aligned}$$

In layman's terms, we are saying that if (and only if) an inequality is satisfied for the minimum value of its LHS (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:

$$l_i(s) = G_i\tau + G_i^{np}s + g_i,$$

where  $G_i^{np} = [G_i^{n\top} \ G_i^{p\top}]$ ,  $G_i^p \in \mathbb{R}^n$  contains only the positive elements of  $G_i$  and zero elsewhere, while  $G_i^n \in \mathbb{R}^n$  contains only its negative elements. The rationale behind this (apparently magic) simplification is that we do not check that an inequality is satisfied for all the values inside the hyper-rectangle: 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. By selecting the negative elements of  $G_i$  for the product with the upper bounds of  $U$  and the positive elements of  $G_i$  for the product with the lower bounds of  $U$ , we are selecting the worst corner. This allows us to reformulate (14) as a convex optimization problem with linear constraints whose solution will be our robustness measure:

$$\begin{aligned} R_{rect}(\tau) = \underset{s}{\text{minimize}} \quad & - \sum_{i=1}^m \log(F_i(s_i) - F_i(s_{i+n})) \\ \text{subject to} \quad & G\tau + G^{np}s + g \geq 0 \\ & s_i > s_{i+n} \quad i = 1 \dots n, \end{aligned} \quad (15)$$

where we constrained each upper bound of  $U(s)$  to be greater than the lower bound to be able to evaluate the logarithm<sup>6</sup>.

### B. Robust TSID

Now that we got rid of the infinite number of constraints, we need to understand how we can minimize  $p_{rect}(\tau)$  with respect to  $\tau$ ,  $p_{rect}(\tau)$  being the solution of an optimization problem. The answer is surprisingly simple: we perform both

<sup>6</sup>In practice we set  $s_i \geq s_{i+n} + \epsilon$ , where  $\epsilon$  is a small positive constant, e.g.  $10^{-6}$ .

optimizations at the same time, which gives us:

$$\begin{aligned} \underset{\tau, s}{\text{minimize}} \quad & \frac{1}{2} \|D\tau - d\|^2 - w \sum_{i=1}^m \log(F_i(s_i) - F_i(s_{i+n})) \\ \text{subject to} \quad & G\tau + G^{np}s + g \geq 0 \\ & s_i > s_{i+n} \quad i = 1 \dots n \end{aligned} \quad (16)$$

This means that we look at the same time for the solution of the original problem  $\tau$  and for the ‘‘best’’ enclosed hyper-rectangle. This is again a convex optimization problem with linear constraints. To solve it with a Newton's method, we need the gradient and the Hessian of the cost function  $c(\tau, s)$ , which we can compute in a few microseconds thanks to their analytical expressions (see Appendix).

### C. Single-Parameter Hyper-Rectangle $p_{box}$

If we parametrize  $U(s)$  with a single variable  $s \in \mathbb{R}$  we can then get a remarkable simplification in the resulting optimization problem. Let us define  $U(s)$  as:

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

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  $k^\top = [\sigma_1 \ \dots \ \sigma_n]$ . The robustness measure then becomes the solution of this linear program:

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

where  $|G|$  is a matrix containing all the absolute values of the elements of  $G$ . This simplification is possible thanks to the fact that maximizing the probability over the hyper-rectangle is equivalent to maximizing its volume, which is in turn equivalent to maximizing  $s$ . The robust TSID optimization problem is then a simple Quadratic Program:

$$\begin{aligned} \underset{\tau, s}{\text{minimize}} \quad & \|D\tau - d\|^2 - ws \\ \text{subject to} \quad & G\tau - |G|ks + g \geq 0 \end{aligned} \quad (17)$$

To solve this problem we can use a standard QP solver, as is usually the case for the classic TSID.

## VI. RELATED WORKS

Considering the robustness of the control scheme is a well-identified problem, but it remains largely unanswered. The classic approach is to consider additive noise disturbing the objective function. When the noise is Gaussian and the objective is quadratic, we can solve the problem with the linear-quadratic-Gaussian regulator (LQG), which has been exploited for animating complex robots [17, 18, 19]. However, considering uncertainties only in the objective cannot ensure any margin [20] and is therefore of limited interest when the system is constrained by strict limitations. Modeling noise in the inequality constraints leads to resolution times that are too long for robot control. The attempts to address this issue, mostly in trajectory optimization, have led to results of limited

practical interest. In [21, 22], the robustness is expressed by considering the stability of the resulting trajectories inside a locomotion cycle. In [23], several variations of the uncertain parameters are simultaneously considered, resulting in a trajectory that is valid for any of the variations. Despite their clear interest, none of these approaches could nowadays be used for online control.

TSID is strongly influenced by parameters that are difficult to identify (e.g. body inertias, joint frictions). Adaptive control [24] and time-delay estimation [25] try to correct online the estimation of the major dynamic effects, while other schemes do not make use of the dynamic model of the robot, and therefore are robust to misestimations [26]. However, these schemes cannot consider strict constraints: in other words, they cannot provide any margin or guarantee on the quality of the control.

In most practical cases, the objective assigned to the robot leaves redundancy in the choice of the decision variables. We suggested to use this redundancy to select the most robust solution. Originally, TSID [27] selects the motion variables answering to the principle of least action, i.e. the closest to the free fall [28], which neither stabilize the system nor avoid violation of the constraints. A common heuristic is to impose a secondary objective to keep the robot posture close to a reference one [29]. Even if this heuristic is rarely commented, it tends to keep the movement away from the boundaries (typically, joint limits) and therefore it increases robustness. Few works yet exist on handling the force redundancy, which only sometimes exists (e.g. in humanoid double-support stances, infinitely-many torques produces the same joint accelerations [30]). Once more, a heuristic exists that tends to produce robust contact forces [31].

## VII. 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?
- Is it worth to use robust TSID even in case of unmodeled uncertainties, such as errors in the inertial parameters?

All tests concern the control of the 30-degree-of-freedom humanoid robot HRP-2, which is always standing on its two feet in plane rigid contact with the environment. Table I lists all the simulation parameters. If not stated otherwise, the inequality constraints are: 22 constraints for the contact forces at the two feet (friction cones, centers of pressure, minimum normal force), 60 constraints for the torque limits (upper and lower bounds), and 60 constraints for the joint acceleration limits (these include also the joint position and velocity bounds). For the simulations we integrated the equations of motion of the system with a first-order Euler scheme with fixed time step  $dt$ . We assumed that the joint torques were affected by Gaussian noise with standard deviation  $\sigma$  proportional to the relative maximum torque  $\tau^{max}$ .

TABLE I: Simulation parameters.

Symbol	Meaning	Value
$dt$	Simulation time step	0.005 s
$\dot{v}_j^{max}$	Max joint acceleration	100 rad s <sup>-2</sup>
$v_j^{max}$	Max joint velocity	0.5 rad s <sup>-1</sup>
$\mu$	Force friction coefficient	0.3
$\mu_m$	Moment friction coefficient	0.1
$f_{min}$	Minimum normal force	1 N
$\sigma$	Torque standard deviation	0.003 $\tau^{max}$
$w$	Robustness weight	10 <sup>-5</sup> (Test 1,2) / 1 (others)
$w_{post}$	Posture weight	10 <sup>-3</sup>
$w_{force}$	Force regularization weight	[1 1 10 <sup>-3</sup> 2 2 2]
$K_p^{post}$	Posture proportional gain	diag([1 ... 1])
$K_d^{post}$	Posture derivative gain	diag([1 ... 1])
$K_p^{com}$	CoM proportional gain	diag([100 100 100])
$K_d^{com}$	CoM derivative gain	diag([20 20 20])
$\epsilon_{accuracy}$	Nonlinear solver accuracy	10 <sup>-6</sup>
$t_{max}$	Max computation time	0.8 ms

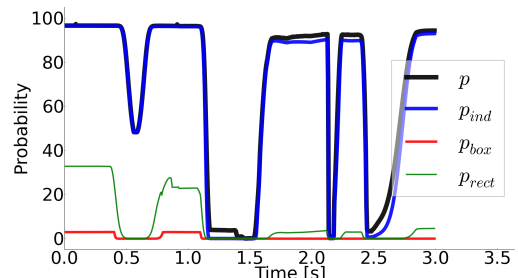


Fig. 4: Test 1. Probability of the inequalities of the TSID control problem computed by the three approximations proposed in this paper.

### A. 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.  $q, v$ ) by controlling the motion of the CoM of the robot with classic TSID (11). 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. 4). We then compared  $p(\tau)$  with the three approximations  $p_{ind}$ ,  $p_{rect}$  and  $p_{box}$ . While  $p_{ind}$  is always quite close to  $p$ ,  $p_{rect}$  and  $p_{box}$  are often far below  $p$ . The average error  $|p - p_{...}|$  is 2.6 for  $p_{ind}$ , 60 for  $p_{rect}$  and 68 for  $p_{box}$ . However, as we will see in the next tests, even maximizing an inaccurate approximation of  $p$  can lead to great improvements in robustness.

### B. Test 2 — Comparing Robustness

The goal of this second test is to compare different TSID formulations in terms of robustness of the inequality constraints. To do so, we used the same state trajectory generated

TABLE II: Results of Test 2. For each formulation we report the average values over 700 tests.

Formulation	Classic TSID	Classic TSID	Robust TSID	Robust TSID	Robust TSID
	$w_f = 0$	$w_f > 0$	$p_{ind}$	$p_{box}$	$p_{rect}$
Probability $p(\tau)$	25.1	27.4	75.7	66.5	72.3
Force prob.	28.2	31.0	86.0	72.2	81.1
Joint-accelerat. prob.	85.2	85.2	85.2	85.3	85.9
Joint-torque prob.	100	100	100	100	100
Active inequalities	3.37	2.83	0.6	0.95	0.01
Iterations	1.06	1.05	2.06	1.11	3.96
Comput. time [ms]	0.23	0.19	0.31	0.2	5.62

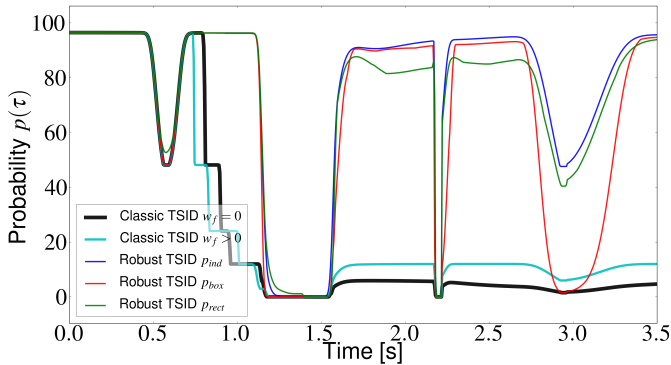


Fig. 5: Test 2. Probability of feasibility  $p(\tau)$  over time of the control determined by different TSID formulations.

for Test 1. For each state we solved the associated control problems using five TSID formulations and we measured  $p(\tau)$ . The main task was to follow a minimum-jerk reference 3D trajectory with the CoM while maintaining as close as possible to the initial joint posture<sup>7</sup>. For the robust formulations, the secondary task was to optimize robustness, i.e. to minimize  $R(\tau)$ . We summed the two cost functions together using a weight  $w$  for the robustness task. For the classic TSID formulation we also added the force-regularization task proposed in [33] (which aims to minimize tangential forces and moments). Finally the task was:

$$\begin{bmatrix} J_{com} & 0 & 0 \\ w_{post}S & 0 & 0 \\ 0 & w_f A_{force} & 0 \end{bmatrix} \begin{bmatrix} \dot{v} \\ f \\ \tau \end{bmatrix} = \begin{bmatrix} \ddot{x}_{com}^{des} - \dot{J}_{com}v \\ w_{post}\ddot{q}_j^{des} \\ 0 \end{bmatrix},$$

where:

$$\begin{aligned} A_{force} &= \text{diag}([w_{force} \quad w_{force}]) \\ \ddot{x}_{com}^{des} &= \ddot{x}_{com}^{ref} + K_p^{com}(x_{com}^{ref} - x_{com}) + K_d^{com}(\dot{x}_{com}^{ref} - \dot{x}_{com}) \\ \ddot{q}_j^{des} &= K_p^{post}(q_j^{ref} - q_j) - K_d^{post}\dot{q}_j \end{aligned}$$

These are the five formulations that we tested:

- Classic TSID with  $w_f = 0$  (11)
- Classic TSID with  $w_f = 10^{-4}$  (11)
- Robust TSID with  $p_{ind}$  approximation (13)
- Robust TSID with  $p_{rect}$  approximation (16)
- Robust TSID with  $p_{box}$  approximation (17)

<sup>7</sup>The postural task is necessary to maintain stability [32].

TABLE III: Results of Test 3. We report the average values for each tested TSID formulation.

Formulation	Classic TSID	Robust TSID	Robust TSID
	$w_f > 0$	$p_{ind}$	$p_{box}$
Probability $p(\tau)$	26.04	99.23	82.96
Force probability	65.29	99.99	90.97
Joint-acceleration probability	67.88	99.25	83.68
Joint-torque probability	100	100	100
Capture-point probability	59.06	99.98	96.11
Active inequalities	1.83	0	0.76
Mean comput. time [ms]	0.18	0.33	0.24
Max comput. time [ms]	1.8	1.06	1.94
Mean comput. time (no warm start) [ms]	10.06	9.36	11.55
Max comput. time (no warm start) [ms]	22.7	23.89	40.57

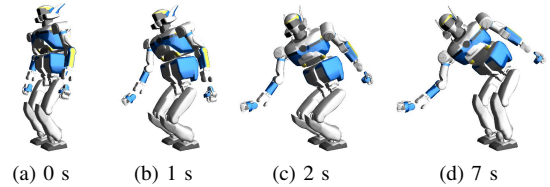


Fig. 6: Test 3. Snapshots of a reaching task.

Table II and Fig. 5 report the results. In terms of probability to satisfy the inequalities the three robust formulations greatly outperform the two classic formulations. The force regularization (i.e.  $w_f > 0$ ) slightly improves the overall probability. Among the robust formulations, the optimization of  $p_{box}$  leads to a probability slightly lower than the others, 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 ( $< 10^{-3}$ ). Interestingly, the  $p_{rect}$  formulation took on average more iterations and computation time (about  $\times 20$ ) than the  $p_{ind}$  formulation, while achieving similar results<sup>8</sup>.

### C. Test 3 — Comparing Computation Time and Overall Behavior

The previous test focused on comparing the robustness of the solutions found by different TSID formulations for the same robot state. Now we focus instead on the robustness of the whole generated trajectory and the computation time. The main task was to reach a point far in front with the right hand (see Fig. 6). To avoid falling we constrained the capture point of the robot to lie inside the support polygon [34]. We ran a simulation for each solver, in which we used its solution (i.e.  $\tau$ ) to simulate the system and get its new state.

Table III reports the average probabilities  $p(\tau)$  for each test. Fig. 7 shows that robust TSID with  $p_{ind}$  approximation leads always to a higher probability than the others. Robust TSID with  $p_{box}$  approximation leads to a higher probability than classic TSID. The task-tracking error is kept small by all the

<sup>8</sup>We could reduce the computation time of the  $p_{rect}$  formulation by exploiting the sparsity of its Hessian.



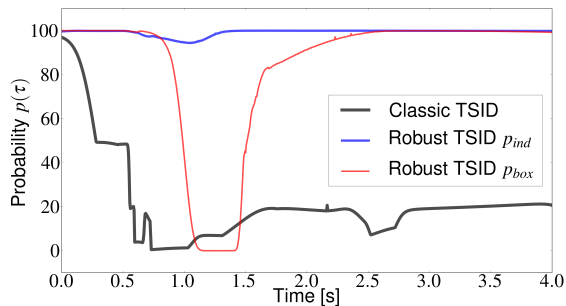


Fig. 7: Test 3. Probability to satisfy all the inequality constraints  $p(\tau)$  using classic and robust TSID.

formulations, so the gains in robustness do not lead to a loss in performances.

#### D. Warm-start

To speed-up the resolution we exploited the warm-start capabilities of *qpOases* [35], the active-set QP solver that we used. For the nonlinear problem we implemented a Sequential Quadratic Programming (SQP) algorithm [36]. We initialized the SQP search with the last solution, which most of the times led to convergence in one Newton's iteration. We used a line-search algorithm that enforces strong Wolfe conditions [36]. The algorithm stopped as soon as the squared newton decrement [16] was less than the desired accuracy ( $\Delta x_{newton}^T H \Delta x_{newton} < 2\epsilon_{accuracy}$ ) or the computation time exceeded the 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.

#### E. Test 4 — Robustness to Noise Probability Distribution

This test verifies whether we can benefit from robust TSID when the noise probability distribution is not known exactly. We repeated the same motion of Test 3 using different values

TABLE IV: Results of Test 4. Probability to satisfy the constraints  $p(\tau)$  when the noise distribution used in the controller differs from the real one  $\sigma = 0.003\tau^{max}$ .

Standard deviation $\sigma$ used in controller	Robust TSID	Robust TSID
	$p_{ind}$	$p_{box}$
0.001 $\tau^{max}$	84.23	86.42
0.002 $\tau^{max}$	98.05	79.3
0.003 $\tau^{max}$	99.23	82.96
0.005 $\tau^{max}$	95.93	84.73
0.006 $\tau^{max}$	87.27	85.89

TABLE V: Results of Test 5. Probability to satisfy the constraints  $p(\tau)$  when the robot model used in the controller differs from the real one.

Robot Model	Classic TSID		Robust TSID $p_{ind}$	
	$p(\tau)$	Ineq. Violation	$p(\tau)$	Ineq. Violation
Perfect	60.29	No	98.92	No
Corrupted 1	53.38	Yes	86.22	Yes
Corrupted 2	55.0	Yes	98.11	No
Corrupted 3	69.05	No	98.94	No
Corrupted 4	54.71	Yes	98.54	No
Corrupted 5	69.24	No	98.94	No
Corrupted 6	64.64	Yes	98.93	No
Corrupted 7	51.74	Yes	58.06	Yes
Corrupted 8	51.74	Yes	59.62	Yes
Corrupted 9	69.16	No	98.94	No
Corrupted 10	54.56	Yes	97.59	No

for the noise standard deviations in the controller. We then computed the probability to satisfy the constraints  $p(\tau)$  using the real standard deviations. As shown in Table IV the gain in robustness with respect to the classic TSID (which led to  $p(\tau) = 26.04$ ) is large regardless of the value of  $\sigma$ .

#### F. Test 5 — Robustness to Inertial Parameters

We generated ten models of HRP-2, differing from each other only for the link masses (each mass was multiplied by a random number between 0.97 and 1.03). We performed the same reaching task described in Section VII-C with classic TSID and  $p_{ind}$  robust TSID using the corrupted models. The results reported in Table V show that robust TSID always led to a higher probability to satisfy the constraints and, most importantly, it violated the deterministic constraints only 3 times, compared to 7 times for classic TSID.

## VIII. CONCLUSION

This paper extended the TSID control framework to take the robustness of the inequality constraints into account. The inclusion of additive random white noise on the joint torques gives rise to a stochastic optimization problem that is too hard to solve in few ms. To overcome this issue we proposed three ways to approximate this problem and we showed through simulations that they lead to large increases in the probability to satisfy the inequality constraints. We also empirically showed that the same holds true even if the uncertainties affecting the system are not the same considered by the controller.

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 [37]) 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.

## ACKNOWLEDGMENTS

This work was supported by Euroc FP7 Grant Agreement 608849 and Entracte ANR grant agreement 13-CORD-002-01. The authors would like to thank Dr. René Henrion for his introductory course on chance constraint optimization.

## REFERENCES

- [1] Twan Koolen, Jesper Smith, Gray Thomas, Sylvain Bertrand, John Carff, Nathan Mertins, Douglas Stephen, Peter Abeles, Johannes Engelsberger, Stephen Mccrory, Jeff Van Egmond, Maarten Griffioen, Marshall Floyd, Samantha Kobus, Nolan Manor, Sami Alsheikh, Daniel Duran, Larry Bunch, Eric Morphis, Luca Colasanto, Khai-long Ho Hoang, Brooke Layton, Peter Neuhaus, Matthew Johnson, and Jerry Pratt. Summary of Team IHMC s Virtual Robotics Challenge Entry. In *Humanoid Robots (Humanoids)*, 2013.
- [2] Salman Faraji, Soha Pouya, Christopher G Atkeson, and Auke Jan 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*, pages 1943–1950, 2014.
- [3] Alexander Herzog, Ludovic Righetti, Felix Grimmering, Peter Pastor, and Stefan Schaal. Balancing experiments on a torque-controlled humanoid with hierarchical inverse dynamics. In *Intelligent Robots and Systems (IROS), 2014 IEEE/RSJ International Conference on*, number Iros, 2014.
- [4] Jun Nakanishi, R. Cory, Michael Mistry, Jan Peters, and Stefan Schaal. Operational Space Control: A Theoretical and Empirical Comparison. *The International Journal of Robotics Research*, 27(6):737–757, June 2008.
- [5] Francesco Nori, Silvio Traversaro, Jorhabib Eljaik, Francesco Romano, Andrea Del Prete, and Daniele Pucci. iCub Whole-Body Control through Force Regulation on Rigid Non-Coplanar Contacts. *Frontiers in Robotics and AI*, 2, 2015.
- [6] Andrea Del Prete, Francesco Nori, Giorgio Metta, and Lorenzo Natale. Prioritized Motion-Force Control of Constrained Fully-Actuated Robots: "Task Space Inverse Dynamics". *Robotics and Autonomous Systems*, 2014.
- [7] Luis Sentis and Oussama Khatib. Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics*, 2(4):505–518, 2005.
- [8] Oscar E Ramos, Nicolas Mansard, Olivier Stasse, Jean-bernard Hayet, and Philippe Soueres. Towards reactive vision-guided walking on rough terrain: an inverse-dynamics based approach. *International Journal of Humanoid Robotics*, 2, 2014.
- [9] Adrien Escande, Nicolas Mansard, and Pierre-Brice Wieber. Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *International Journal of Robotics Research*, 2014.
- [10] René Henron. Introduction to Chance Constrained Programming. *Tutorial paper for the Stochastic Programming Community*, 2004.
- [11] Alan Genz. Numerical computation of multivariate normal probabilities. *Journal of computational and graphical statistics*, 1(2):140–149, 1992.
- [12] Layale Saab, Oscar Ramos, Nicolas Mansard, Philippe Souères, and Jean-Yves Fourquet. Dynamic Whole-Body Motion Generation Under Rigid Contacts and Other Unilateral Constraints. *IEEE Trans. on Robotics*, 29(2), April 2013.
- [13] Sebastien Rubrecht, Vincent Padois, Philippe Bidaud, and Michel 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.
- [14] Andrea Del Prete, Nicolas Mansard, Francesco Nori, Giorgio Metta, and Lorenzo Natale. Partial Force Control of Constrained Floating-Base Robots. In *Intelligent Robots and Systems (IROS), IEEE International Conference on*, 2014.
- [15] H. Vazquez-Leal. High accurate simple approximation of normal distribution integral. *Mathematical problems in engineering*, 2012.
- [16] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*, volume 98. 2004.
- [17] W. Li and E. Todorov. Iterative linearization methods for approximately optimal control and estimation of non-linear stochastic system. *International Journal of Control*, 80(9):1439–1453, September 2007.
- [18] K. Rawlik, M. Toussaint, and S. Vijayakumar. On stochastic optimal control and reinforcement learning by approximate inference. *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, 2013.
- [19] Andrea Del Prete, Francesco Romano, Lorenzo Natale, Giorgio Metta, Giulio Sandini, and Francesco Nori. Prioritized Optimal Control. In *Robotics and Automation (ICRA), IEEE International Conference on*, 2014.
- [20] J. Doyle. Guaranteed margins for LQG regulators. *IEEE Transactions on Automatic Control*, 23(4):756–757, August 1978.
- [21] Katja Mombaur, Richard Longman, Hans Georg Bock, and Johannes Schlöder. Open-loop stable running. *Robotica*, 23(1):21–33, January 2005.
- [22] E. R. Westervelt, J. W. Grizzle, and C. Chevallereau. *Feedback control of dynamic bipedal robot locomotion*. 2007.
- [23] Igor Mordatch, Kendall Lowrey, and Emanuel Todorov. Ensemble-CIO : Full-Body Dynamic Motion Planning that Transfers to Physical Humanoids. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.
- [24] R. Kelly and R. Carelli. On adaptive impedance control of robot manipulators. *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, 1989.
- [25] 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*, 55(1), 2008.
- [26] Alexander Dietrich, Christian Ott, and Alin Albu-

- Schaffer. Multi-objective compliance control of redundant manipulators: Hierarchy, control, and stability. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3043–3050, Tokyo, Japan, November 2013.
- [27] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, February 1987.
- [28] Herman Bruyninckx and Oussama Khatib. Gauss’ principle and the dynamics of redundant and constrained manipulators. In *Robotics and Automation, 2000. Proceedings. ICRA’00. IEEE International Conference on.*, 2000.
- [29] Luis Sentis. *Synthesis and control of whole-body behaviors in humanoid systems*. PhD thesis, Stanford University, 2007.
- [30] Luis Sentis, Jaeheung Park, and Oussama Khatib. Compliant control of multicontact and center-of-mass behaviors in humanoid robots. *Robotics, IEEE Transactions on*, 2010.
- [31] Ludovic Righetti, Jonas Buchli, Michael Mistry, and Stefan 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*, pages 580–587, Singapore, 2010. World Scientific Publishing Co. Pte. Ltd.
- [32] Jan Peters, Michael Mistry, Firdaus E. Udawadia, Jun Nakanishi, and Stefan Schaal. A unifying framework for robot control with redundant DOFs. *Autonomous Robots*, 24(1):1–12, October 2007.
- [33] Ludovic Righetti, Jonas Buchli, Michael Mistry, Mrinal Kalakrishnan, and Stefan Schaal. Optimal distribution of contact forces with inverse dynamics control. *The International Journal of Robotics Research*, (January), January 2013.
- [34] Oscar E Ramos, Nicolas Mansard, and Philippe 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.
- [35] H. J. Ferreau, C. Kirches, and A. Potschka. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 2013.
- [36] Stephen J. Wright and Jorge Nocedal. *Numerical optimization*. 1999.
- [37] Andrea Del Prete, Nicolas Mansard, Oscar E. Ramos, Olivier Stasse, and Francesco Nori. Implementing Torque Control with High-Ratio Gear Boxes and without Joint-Torque Sensors. *International Journal of Humanoid Robotics (submitted to)*, 2015.