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

Andrea del Prete, Nicolas Mansard

Addressing Constraint Robustness 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, sensors noise, model uncertainties and delays. 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 joints 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 [15, 7, 11, 19]. This technique offers a number of attractive features: it is theoretically sound, it works well in simulation [30, 23], 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 joints 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[1] 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 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 in TSID greatly increases the chances to have feasible control trajectories for a system subject to uncertainties (even when these uncertainties are not exactly 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 the control of a real robot.

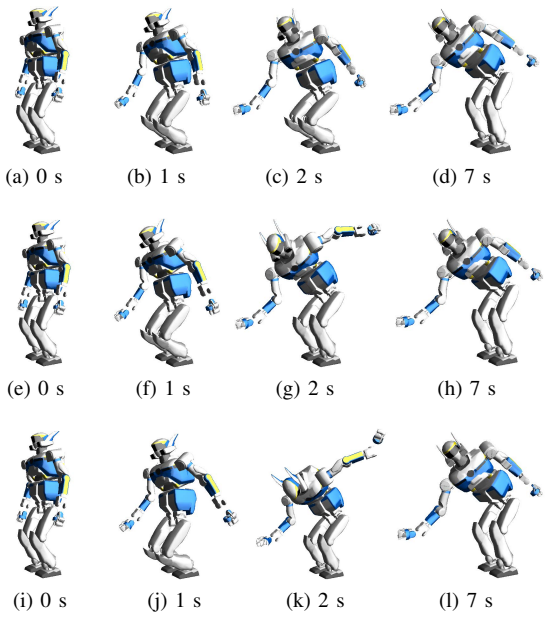


Fig. 1: Snapshots of a reaching task executed with classic TSID (first line), robust TSID with p_{ind} approximation (second line) and robust TSID with p_{box} approximation.

II. From Deterministic to Stochastic Optimization

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

$$\begin{aligned} \min_{x} \quad & ||Ax - a||^2 \\ \text{s.t.} \quad & Bx + b \geq 0 \end{aligned} \tag{1}$$

[1] In a stochastic optimization problem some of the variables are random and follow a known probability distribution

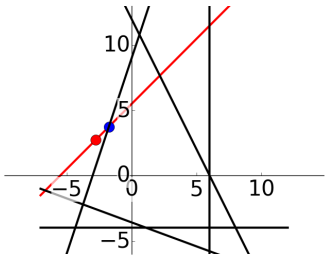


Fig. 2: 2D example of inequality-constrained least-square problem solved by an active-set method. The red line represents the set of solutions of $Ax = a$, the black lines represent the inequality constraints $Bx + b \geq 0$, the red dot is the least-norm solution of $Ax = a$ and the blue dot is the solution of (1) determined by an active-set method.

In the next section we will see how this relates to the problem of controlling a robot with TSID. 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 [6], mainly because they can be easily warm-started [2] (contrary to interior-point methods). Take for instance the 2D example depicted in Fig. 2. The active-set algorithm first computes a solution of the unconstrained problem (i.e. red dot). Since this solution 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. 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[3] $\Sigma = \text{diag}(\begin{bmatrix}\sigma_1^2 & \dots & \sigma_n^2\end{bmatrix})$) affecting the problem's variable x, that is:

$$\begin{aligned} \min_{x} \quad & ||A(x+e) - a||^2 \\ \text{s.t.} \quad & B(x+e) + b \geq 0 \end{aligned} \tag{2}$$

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. We typically replace them with their probability to be satisfied [10]:

$$p(x) = \text{P}(B(x+e) + b > 0) \tag{3}$$

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

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

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

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.

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

$$\begin{aligned} \min_{x} \quad & ||Ax - a||^2 \\ \text{s.t.} \quad & R(x) < R_{min} \end{aligned} \tag{5}$$

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:

$$\begin{aligned} \min_{x} \quad & ||Ax - a||^2 + wR(x) \\ \text{s.t.} \quad & Bx + b \geq 0 \end{aligned} \tag{6}$$

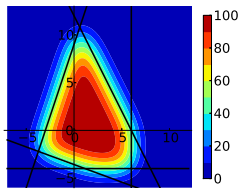
where $w \in \mathbb{R}$ weighs the importance of robustness with respect to cost. Here, we kept the deterministic inequalities to prevent the solution from violating them in favor of minimizing the cost (which may happen if w is not large enough).

Regardless of which of these three formulations we decide to use, 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, so we can only resort to computationally-expensive numerical techniques [9]. 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.

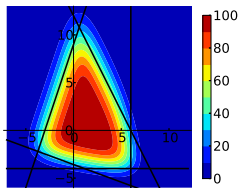
[2]Warm-starting the resolution of an optimization problem consists in exploiting the solution of a similar problem (which was already computed) to speed-up the computation.

[3]All the results can be easily 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).

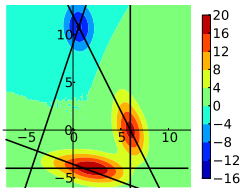
[4]For instance, rather than maximizing a probability we maximize its logarithm because it is a concave function.



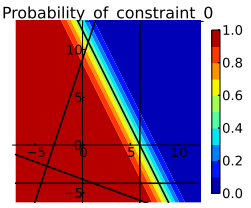
(a) Joint inequalities probability $p(x) = \mathrm{P}(B(x+e)+b > 0)$.



(b) Individual inequalities probability $p_{ind}(x) = \prod_{i=1}^{m} \mathrm{P}(B_i(x+e)+b_i \geq 0)$.



(c) Difference between Fig. 3a and 3b, that is $p(x) - p_{ind}(x)$; max error 20%, mean error 1.8%.



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

Fig. 3: Probability of satisfying a set of 5 stochastic inequalities $B(x+e)+b \geq 0$, where $e_i \sim \mathcal{N}(0, 1.44)$ for $i = 1, 2$.

B. Approximation 1 — Individual Constraints p_{ind}

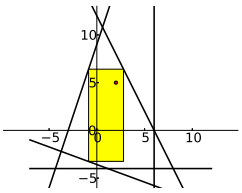
The first way to simplify (3) is by considering the probabilities of the single inequalities rather than the probability of all of them:

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

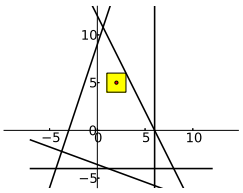
where B_i is the i-th row of B. When e is Gaussian, this is equivalent to neglecting the off-diagonal terms of the covariance matrix of e_B. This approximation fastn to resolution of (3) because we only need m univariate — rather than one multivariate — cdfs. To get a better intuition of why $p_{ind}(x)$ is a good approximation of $p(x)$ let us look at a simple 2D example. Fig. 3a 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. 3b shows the approximated probability $p_{ind}(x)$ obtained with (7), 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 smaller 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.

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

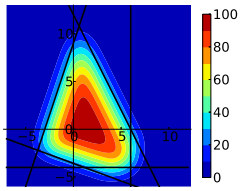
In the first approximation we exploited the fact that we can easily compute the probability of a single inequality. Another



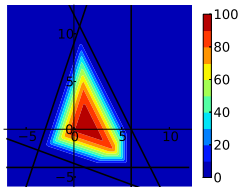
(a) Hyper-rectangle approximation.



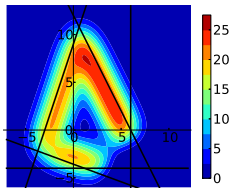
(b) Box approximation.



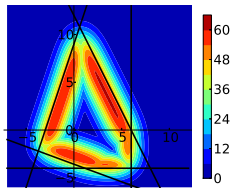
(c) Hyper-rectangle probability p_{rect}.



(d) Cube probability p_{box}.



(e) Hyper-rectangle probability error $(p(x) - p_{rect}(x))$, max error 25%, mean error 4.7%.



(f) Cube probability error $(p(x) - p_{box}(x))$, max error 61%, mean error 12.4%.

Fig. 4: Probability of satisfying a set of 5 stochastic inequalities $B(x+e)+b \geq 0$, where $e_i \sim \mathcal{N}(0, 1.44)$ for $i = 1, 2$.

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, we can compute the joint probability as 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:

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

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

$$\begin{aligned} p_{rect}(x) = \max_{s} \quad & \mathrm{P}((x+e) \in U(s)) \\ \text{s.t.} \quad & Bz + b \geq 0 \qquad \forall z \in U(s), \end{aligned} \tag{9}$$

where the (infinite) constraints ensure that $U(s)$ is enclosed in the feasible set. Fig. 4 shows the hyper-rectangle maximizing the probability for different parametrizations s. In Fig. 4a $U(s)$ is a general hyper-rectangle and hence, we need $2n$ values

to represent it. In Fig. 4b $U(s)$ 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 parametrizations trade off the quality of the approximation for its simplicity. Fig. 4c and 4d show the value of p_{rect} and p_{box} over the solution space, whereas Fig. 4e and 4f show the approximation errors $(p(x) - p_{rect}(x))$ and $(p(x) - p_{box}(x))$. While this approximation may seem much coarser than the first one, in Section VII we will show empirically that it performs rather 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 then 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. We use here the formulation of [27], written as an optimization problem of τ, f and $\dot{v}$:

$$\begin{aligned} \min_{y} \quad & ||Ay - a||^2 \\ \text{s.t.} \quad & By + b \geq 0 \\ & \begin{bmatrix} J_c & 0 & 0 \\ M & -J_c^\top & -S^\top \end{bmatrix} \underbrace{\begin{bmatrix} \dot{v} \\ f \\ \tau \end{bmatrix}}_{y} = \begin{bmatrix} -\dot{J}_c v \\ -h \end{bmatrix}, \end{aligned} \tag{10}$$

where $\dot{v}$ are the base and joints accelerations, f are the contact forces, τ are the joints 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 joints acceleration limits. The bounds of the joints positions and velocities are typically converted into joints accelerations bounds [20]. The cost function represents the error of the task, which is typically a linear function of $\dot{v}$ (i.e. a task-space acceleration):

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

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 joints 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 τ alone by expressing $\dot{v}$ and f as functions of τ [5]:

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

[5]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 [3].

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:

$$\begin{aligned} \min_{\tau} \quad & ||D\tau - d||^2 \\ \text{s.t.} \quad & G\tau + g \geq 0, \end{aligned} \tag{11}$$

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:

$$\begin{aligned} \min_{\tau} \quad & ||D\tau - d||^2 + wR(\tau) \\ \text{s.t.} \quad & G\tau + g \geq 0 \end{aligned} \tag{12}$$

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:

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

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 [31] (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 $\sigma_{G_i} = \sigma_i^2 G_i G_i^\top$. 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 $F_{G_i}(.)$ 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 [1] (see the Appendix for the expressions of the gradient and the Hessian of R_{ind}). The final robust TSID problem is then a nonlinear convex optimization:

$$\begin{aligned} \min_{\tau} \quad & ||D\tau - d||^2 - w \sum_{i=1}^{m} \log F_{G_i}(G_i\tau + g_i) \\ \text{s.t.} \quad & G\tau + g \geq 0 \end{aligned} \tag{13}$$

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:

$$\begin{aligned} p_{rect}(\tau) = \max_{s} \quad & \mathrm{P}(e \in U(s)) \\ \text{s.t.} \quad & G(\tau + z) + g \geq 0, \quad \forall z \in U(s) \end{aligned} \tag{14}$$

First of all, we will define $U(s)$ for the general case, and then we will show how things simplify for the other cases. 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 Σ is diagonal results in this simplification:

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

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:

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

where l_i is the solution of an optimization problem:

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

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:

$$G_i^{np} = \begin{bmatrix} G_i^n & G_i^p \end{bmatrix}$$

$G_i^p \in \mathbb{R}^n$ 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:

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

where we explicitly constrained the upper bounds of $U(s)$ to be greater than its relative lower bounds in order to be able to evaluate the logarithm[6].

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 τ, $p_{rect}(\tau)$ being the solution of an optimization problem. The answer is surprisingly simple: we perform both optimizations at the same time, which gives us:

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

This means that we look at the same time for the solution of the original problem τ 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

[6]In practice we set $s_i \geq s_{i+n} + \epsilon$, with ϵ being a small positive constant, such as 10^{-6}.

have always set $k^\top = \begin{bmatrix}\sigma_1 & \dots & \sigma_n\end{bmatrix}$. The robustness measure then becomes the solution of this linear program:

$$R_{box}(\tau) = \min_{s} \quad -s$$
$$\text{s.t.} \quad G\tau - |G|ks + g \geq 0,$$

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:

$$\begin{aligned} \min_{\tau,s} \quad & ||D\tau - d||^2 - ws \\ \text{s.t.} \quad & G\tau - |G|ks + g > 0 \end{aligned} \tag{17}$$

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 regulators (LQG), which has been exploited for animating complex nonlinear robots [16, 24]. However, considering uncertainties only in the objective cannot ensure any margin [5] 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 [17, 32], the robustness is expressed by considering the stability of the resulting trajectories inside a locomotion cycle. In [18], 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, joints frictions). Adaptive control [13] or time-delay estimation [12] 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 [4]. 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 [14] selects the motion variables answering to the principle of least action, i.e. the closest to the free fall [2], which neither stabilizes the system nor enables it to avoid violating 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 joints accelerations [28]). Once more, a heuristic exists that tends to produce robust contact forces [25].

VII. Simulations

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

- What kind of improvements 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?

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. All the parameters of the simulations are listed in Table I. If not stated otherwise, the inequality constraints are:

- 22 constraints for the contact forces at the two feet: friction cones (with friction coefficients μ and μ_m, center of pressure, minimum normal force f_{min}
- 60 constraints for the torque limits: upper and lower bounds
- 60 constraints for the joints accelerations limits $\dot{v}_j^{max}$: upper and lower bounds (these include also the joints positions and velocities bounds v_j^{max})

TABLE I: Simulation parameters.

Symbol	Meaning	Value
dt	Simulation time step	0.005 s
$\dot{v}_j^{max}$	Max joint acceleration	$100 rads^{-2}$
v_j^{max}	Max joint velocity	$0.5 rads^{-1}$
μ	Force friction coefficient	0.3
μ_m	Moment friction coefficient	0.1
f_{min}	Minimum normal force	1 N
σ	Torque standard deviation	$0.003\tau^{max}$
w	Robustness weight	10^{-5} (Test 1,2) / 1 (Test 3,4)
w_{post}	Posture weight	10^{-3}
w_{force}	Force regularization weight	$\begin{bmatrix}1 & 1 & 10^{-3} & 2 & 2 & 2\end{bmatrix}$
K_p^{post}	Posture proportional gain	$\text{diag}(\begin{bmatrix}1 & \dots & 1\end{bmatrix})$
K_d^{post}	Posture derivative gain	$\text{diag}(\begin{bmatrix}1 & \dots & 1\end{bmatrix})$
K_p^{com}	CoM proportional gain	$\text{diag}(\begin{bmatrix}100 & 100 & 100\end{bmatrix})$
K_d^{com}	CoM derivative gain	$\text{diag}(\begin{bmatrix}20 & 20 & 20\end{bmatrix})$
$\epsilon_{accuracy}$	Nonlinear solver accuracy	10^{-6}
t_{max}	Max computation time	0.8 ms

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 joints torques were affected by a Gaussian noise with standard deviation σ proportional to the relative maximum torque τ^{max}.

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 joints 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. 5). We then compared $p(\tau)$ with

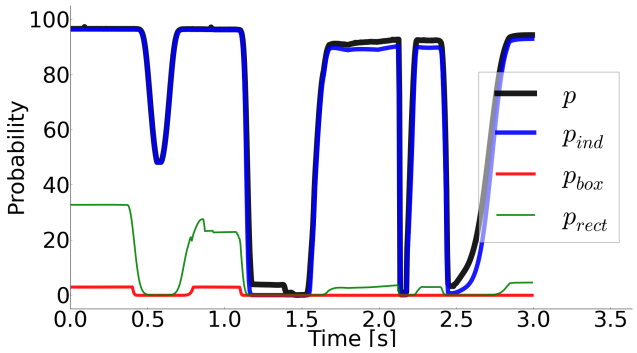


Fig. 5: Test 1. Probability of the inequalities of the TSID control problem computed by the three approximations proposed in this paper.

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, we will see in the next tests that even by maximizing an inaccurate approximation of p we can get 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 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[7]. 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 [26] (which aims to minimize tangential forces and moments). Finally the task was:

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

[7]The postural task is necessary to maintain stability [21].

where:

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

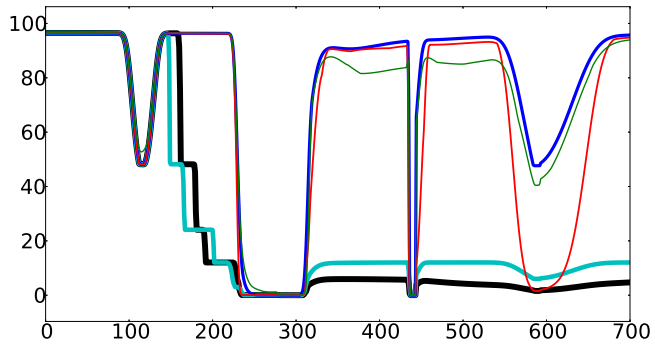
These are the five formulations that we tested:

- 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)

Table II reports the results. In terms of probability to satisfy

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}	Robust TSID p_{rect}
Ineq. prob.	25.1	27.4	75.7	66.5	72.3
Forces ineq. prob.	28.2	31.0	86.0	72.2	81.1
Joints accelerations ineq. prob.	85.2	85.2	85.2	85.3	85.9
Torques ineq. 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



(a) Probability of feasibility $p(\tau)$ over time.

Fig. 6: Test 2. Comparing the robustness of the control determined by different TSID formulations: classic TSID (black), classic TSID with $w_f > 0$ (cyan), p_{ind} robust TSID (blue), p_{box} robust TSID (red), p_{rect} robust TSID (green).

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. Looking at the probability of the forces inequalities only, we can see that it is there that robust and classic formulations differ the most. All the formulations lead to small errors for the CoM task ($< 10^{-3}$). Interestingly, the p_{rect} formulation took on average more iterations and

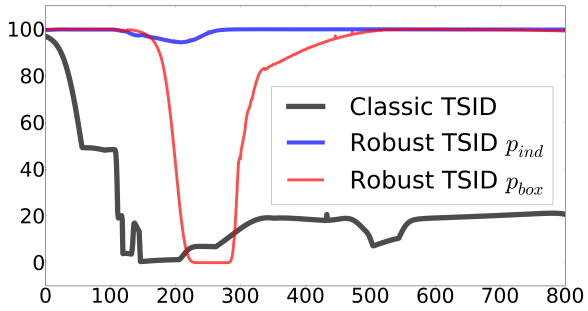
computation time (about $\times 20$) than the p_{ind} formulation, while achieving similar results [8].

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 state of the robot. Now we focus instead on robustness

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

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



(a) Probability to satisfy all the inequality constraints (i.e. $p(\tau)$) over time.

Fig. 7: Test 3. Comparison between the overall robustness of the motion generated by using classic TSID and the two robust TSID formulations p_{ind} and p_{box}.

of the whole generated trajectory and the computation time of the different formulations. The main task was to reach a point far in front with the right hand (see Fig. 1). To avoid falling we added constrained the capture point of the robot to lie inside the support polygon [22]. 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.

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. The solution of robust TSID with p_{box} approximation has a higher probability than that of classic TSID. The task tracking error is kept small by all the formulations, so the gains in robustness do not lead to a loss in performances.

[8]We can probably lower the computation time of the p_{rect} formulation by exploiting the sparsity of its Hessian, but this is outside the scope of this work.

D. Warm-start

To speed-up the resolution we exploited the warm-start capabilities of *qpOases* [8], the active-set QP solver that we used. For the nonlinear problem we implemented a Sequential Quadratic Programming (SQP) algorithm [33]. We initialized the SQP search with the last solution, which most of the times led to convergence in one Newton's iteration. We also used a line-search algorithm that enforces strong Wolfe conditions [33]. We terminated the algorithm as soon as (at least) one of these conditions was satisfied:

- the squared newton decrement [1] was less than the desired accuracy: $\Delta x_{newton}^{\top} H \Delta x_{newton} < 2\epsilon_{accuracy}$
- the computation time exceeded the limit t_{max}

For the computation time, we only measured the time taken by *qpOases*, which means that we neglected the computation of Hessian and gradient of the cost function and the line search. 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 (probably due to the python interface of *qpOases*), the computation time is always below $t_{max} = 0.8$ ms.

E. Test 4 — Robustness to Noise Probability Distribution

In this final test we verified whether we could benefit from the robust TSID formulations when the probability distribution of the noise is not known exactly. We tried repeating the same motion of Test 3 using different values for the standard deviations of the noise. We then computed the probability to satisfy the constraints $p(\tau)$ using the real standard deviations. As shown in Table IV the changes of σ affected the performances of the controller, always negatively for p_{ind}, but sometimes positively for p_{box}. Nonetheless, the gain in robustness with respect to the classic TSID (which led to $p(\tau) = 26.04$) seems to be large regardless of the value of σ.

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 σ used in controller	Robust TSID p_{ind}	Robust TSID 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

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 joints 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 led 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 a real HRP-2 robot 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.

References

[1] Stephen Boyd and L Vandenberghe. *Convex Optimization*, volume 98. 2004.

[2] H Bruyninckx and O Khatib. Gauss' principle and the dynamics of redundant and constrained manipulators. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on.*, 2000.

[3] 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 2014), IEEE International Conference on*, 2014.

[4] 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.

[5] J. Doyle. Guaranteed margins for LQG regulators. *IEEE Transactions on Automatic Control*, 23(4):756–757, August 1978.

[6] Adrien Escande, Nicolas Mansard, and Pierre-Brice Wieber. Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *International Journal of Robotics Research*, 2014.

[7] 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.

[8] HJ Ferreau, C Kirches, and A Potschka. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 2013.

[9] Alan Genz. Numerical computation of multivariate normal probabilities. *Journal of computational and graphical statistics*, 1(2):140–149, 1992.

[10] René Henrion. Introduction to Chance Constrained Programming. *Tutorial paper for the Stochastic Programming Community*, 2004.

[11] Alexander Herzog, Ludovic Righetti, and Felix Grimminger. Experiments with a hierarchical inverse dynamics controller on a torque-controlled humanoid. *arXiv preprint arXiv:1305.2042*, 2013.

[12] M Jin, SH Kang, and PH Chang. Robust compliant motion control of robot with nonlinear friction using time-delay estimation. *Industrial Electronics, IEEE Transactions on*, 55(1), 2008.

[13] R Kelly and R Carelli. On adaptive impedance control of robot manipulators. *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, 1989.

[14] 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.

[15] Twan Koolen, Jesper Smith, Gray Thomas, Sylvain Bertrand, John Carff, Nathan Mertins, Douglas Stephen, Peter Abeles, Johannes Englsberger, 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.

[16] 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.

[17] Katja Mombaur, Richard Longman, Hans Georg Bock, and Johannes Schlöder. Open-loop stable running. *Robotica*, 23(1):21–33, January 2005.

[18] 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.

[19] 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.

[20] Vincent Padois, Philippe Bidaud, Michel De Broissia, and Michel De Broissia Constraints Com. Constraints Compliant Control : constraints compatibility and the displaced configuration approach. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010.

[21] Jan Peters, Michael Mistry, Firdaus E. Udwadia, Jun Nakanishi, and Stefan Schaal. A unifying framework for robot control with redundant DOFs. *Autonomous Robots*, 24(1):1–12, October 2007.

[22] Oscar E Ramos, Nicolas Mansard, Philippe Soueres,

Oscar E Ramos, Nicolas Mansard, Philippe Soueres, and Whole-body Motion Integrating. Whole-body Motion Integrating the Capture Point in the Operational Space Inverse Dynamics Control. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2014.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] L Sentis, J Park, and O Khatib. Compliant control of multicontact and center-of-mass behaviors in humanoid robots. *Robotics, IEEE Transactions on*, 2010.

[29] Luis Sentis. *Synthesis and control of whole-body behaviors in humanoid systems*. PhD thesis, Stanford University, 2007.

[30] 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.

[31] H Vazquez-Leal. High accurate simple approximation of normal distribution integral. *Mathematical problems in engineering*, 2012.

[32] ER Westervelt, JW Grizzle, and C Chevallereau. *Feedback control of dynamic bipedal robot locomotion*. 2007.

[33] SJ Wright and J Nocedal. *Numerical optimization*. 1999.