Robustness to Inertial Parameter Errors for Legged Robots Balancing on Level Ground

Nirmal Giftsun, Andrea Del Prete, Florent Lamiraux

Robustness to Inertial Parameter Errors for Legged Robots Balancing on Level Ground

Nirmal Giftsun[1], Andrea Del Prete[1] and Florent Lamiraux[2]

Laboratory for Analysis and Architecture of Systems, 7 Avenue du Colonel Roche, 31400 Toulouse
ngiftsun@laas.fr[1] adelpret@laas.fr[1] florent@laas.fr[2]

Abstract: Model-based control has become more and more popular in the legged robots community in the last ten years. The key idea is to exploit a model of the system to compute precise motor commands that result in the desired motion. This allows to improve the quality of the motion tracking, while using lower gains, leading so to higher compliance. However, the main flaw of this approach is typically its lack of robustness to modeling errors. In this paper we focus on the robustness of inverse-dynamics control to errors in the inertial parameters of the robot. We assume these parameters to be known, but only with a certain accuracy. We then propose a computationally-efficient optimization-based controller that ensures the balance of the robot despite these uncertainties. We used the proposed controller in simulation to perform different reaching tasks with the HRP-2 humanoid robot, in the presence of various modeling errors. Comparisons against a standard inverse-dynamics controller through hundreds of simulations show the superiority of the proposed controller in ensuring the robot balance.

1 INTRODUCTION

The problem of balancing for real legged robots is still a challenge for the robotics community. Although our understanding of this problem has remarkably improved during the last 15 years, the robustness of the state-of-the-art control algorithms is far from satisfactory. For instance, during the recent DARPA Robotics Challenge Finals (Pratt, 2015), all legged robots have moved extremely cautiously, and, despite that, sometimes they could not avoid falling. Another striking fact is the difference between what robots can do in simulation where they easily perform extremely dynamics tasks and what they can do in the real world where they struggle to execute slow movements in structured environments. The gap between simulation and real world can be explained through countless unmodeled uncertainties affecting these systems, such as poor torque control, model uncertainties, sensor noises and delays.

In our recent work (Del Prete and Mansard, 2016) we have proposed an optimization-based controller that tries to ensure the satisfaction of the physical constraints of the robot (force friction cones, joint acceleration limits and torque limits) despite errors in the joint torque tracking. In this paper we move along the same line, designing a *robust* controller that can

balance a legged robot despite bounded errors in its inertial parameters.

1.1 State of the Art

Even though the problem of robustness is long-standing and well-identified, it remains largely unanswered for legged robots. Some approaches focus exclusively on the stability of the system rather than on the feasibility of the trajectories. For instance, adaptive control (Kelly and Carelli, 1989) and time-delay estimation (Jin et al., 2008) try to estimate and compensate online for the major errors between nominal and real dynamic model. Virtual model control (Pratt et al., 2001) does not rely on the dynamic model of the robot, which ensures robustness to errors in the inertial parameters (Dietrich et al., 2013). The main issue of these schemes is that they do not consider inequality constraints, which makes it hard to implement them on real systems, given the large number of bounds to which they are subject.

Other approaches are based on hand-tunable heuristics. For instance, a common heuristic in Task-Space Inverse Dynamics (TSID) (Del Prete et al., 2015) which we adopt as well is to use a secondary task to keep the robot posture close to a reference one, in order to keep the movements far from the joint lim-

its. Similarly, to avoid slipping/tipping, it was proposed to minimize the contact moments and the tangential contact forces in the null space of the main motion task (Righetti et al., 2010). Yet another common trick during locomotion is to maintain the center of pressure close to the center of the foot (Kajita and Kanehiro, 2003). The robotics literature is filled with these kinds of heuristics, which often are the main reason behind the successful implementations on real platforms. However, these heuristics can not ensure feasibility in the presence of any significant uncertainty and needs ad-hoc tuning depending on the situation.

Finally, another class of works which includes this paper makes use of robust optimization techniques to formulate control and planning problems. Mordatch et al. (Mordatch et al., 2015) considered several perturbed models of a humanoid robot to plan offline a trajectory that is robust to uncertainties, reporting success rate between 80% and 95% on a real platform. Another recent work (Luo and Hauser, 2015) has combined robust and time-scaling optimization to plan trajectories that are robust to bounded errors in friction coefficients and joint accelerations, whose magnitude can be estimated online through iterative learning. Nguyen and Sreenath (Nguyen and Sreenath, 2015) have recently exploited control Lyapunov functions and Quadratic Programs (QPs) to ensure stability despite bounded uncertainties in the linearized system dynamics.

Contrary to (Mordatch et al., 2015; Nguyen and Sreenath, 2015), the uncertainties modeled in this paper affect the parameters of the system, so they could be identified using set-membership identification techniques (Ramdani and Poignet, 2005). The main contribution of this paper is a novel formulation of the capture-point balance constraints, which can be included in the Task-Space Inverse Dynamics optimization problem to balance the robot despite bounded uncertainties in its inertial parameters. Contrary to previous approaches that dealt with uncertainties to inertial parameters, our approach allows us to include inequality constraints in the problem formulation. Thanks to this we can thus account for all the constraints to which legged robots are subject, ensuring the feasibility of the resulting trajectories.

1.2 Paper Overview

In Section 2 we model the uncertainty in the inertial parameters of the robot through polytopes. Then we present the TSID controller with capture-point constraints (Ramos et al., 2014) to ensure the balance of the robot in case of no modeling errors. Section 3 presents an extension of the standard capture-point inequalities that is robust to errors in the inertial parameters. We first formulate the associated robust optimization problem, and then use standard robust-optimization techniques to reformulate it in a tractable form.

Section 4 presents statistical results that compares in simulation the standard and the robust controller in a reaching task with the humanoid robot HRP-2 at different conditions. Regardless of the simulation conditions, our results empirically demonstrate the superiority of the proposed robust controller with respect to standard TSID. Finally, Section 5 draws the conclusions and discusses the future work.

2 INVERSE DYNAMICS IN TASK SPACE WITH CAPTURE POINT BALANCE CONSTRAINTS

To design a controller that is robust to errors in the inertial parameters of the robot we have first to understand how these errors affect the control action. In this section we define the inertial parameters and we present a standard Task-Space Inverse Dynamics controller, which includes balance constraints. Throughout the presentation we explicitly show the dependency of the terms on the inertial parameters, while we omit the dependency on the robot configuration q and velocities v because they are constant values at each time step.

2.1 Inertial Parameters

We define the vector containing the 10 inertial parameters of link i as:

$$\phi_i = (m_i, m_i{}^i c_i, I_i^{xx}, I_i^{xy}, I_i^{xz}, I_i^{yy}, I_i^{yz}, I_i^{zz}),$$

where $m_i \in \mathbb{R}$ is the mass, $c_i \in \mathbb{R}^3$ is the center of mass, $I_i \in \mathbb{R}^{3 \times 3}$ is the 3D rotational inertia matrix. Both c_i and I_i are expressed in the local reference frame of the link. Note that ϕ_i does not contain directly c_i, but it contains only its product with m_i. This is because the robot dynamics can be written in a linear form with respect to this parameterization of the inertial parameters (Traversaro et al., 2015).

Now we can collect the inertial parameters of all the N links of the robot in a single vector:

$$\phi = (\phi_1, \dots, \phi_N)$$

We assume that each link parameters ϕ_i are not known exactly, but we know that they lie inside a polytope

U_i, i.e. $\phi_i \in U_i$. Hence also the vector ϕ lies inside a polytope:

$$\phi \in U = U_1 \times \cdots \times U_N$$

Note that since a polytope can be represented by a set of linear inequalities, the constraint $\phi_i \in U_i$ can be expressed under the form $A_i \phi_i \leq a_i$.

2.2 Task-Space Inverse Dynamics

Now that we defined the inertial parameters and the associated uncertainty polytopes, we can see how these uncertainties affect the controller.

The controller that we consider in this paper is an optimization-based inverse dynamics controller, which has become a standard for the control of legged robots in recent years (Del Prete et al., 2015; Herzog et al., 2016; Sentis and Khatib, 2004; Saab et al., 2011). Various formulations of the TSID optimization problem exist and are often equivalent or similar (Del Prete et al., 2015). We write it here as an optimization problem of the base and joint accelerations $\dot{v} \in \mathbb{R}^{n+6}$, the contact forces $f \in \mathbb{R}^k$, and the joint torques $\tau \in \mathbb{R}^n$ (Saab et al., 2013):

$$\underset{y=(\dot{v},f,\tau)}{\text{minimize}} \quad ||Ay - a||^2$$

$$\text{subject to} \quad \begin{bmatrix} M(\phi) & -J_c^\top & -S^\top \\ J_c & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{v} \\ f \\ \tau \end{bmatrix} = \begin{bmatrix} -h(\phi) \\ -\dot{J}_c v \end{bmatrix}$$

$$|\tau| \leq \tau^{max} \tag{1a}$$

$$v^{min} \leq \dot{v} \leq v^{max} \tag{1b}$$

$$f \in \mathcal{K} \tag{1c}$$

where $J_c \in \mathbb{R}^{k \times (n+6)}$ is the constraint Jacobian, $M \in \mathbb{R}^{(n+6) \times (n+6)}$ is the mass matrix, $h \in \mathbb{R}^{n+6}$ contains the bias forces, $S \in \mathbb{R}^{n \times (n+6)}$ is the selection matrix, $\tau^{max} \in \mathbb{R}^n$ are the maximum joint torques, $v^{min/max} \in \mathbb{R}^{n+6}$ are the acceleration bounds[1], and $\mathcal{K}$ is the force friction cone (which is typically linearized). The cost function represents the error of the task, which is typically an affine function of $\dot{v}$ (i.e. a task-space acceleration):

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

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

---

[1] The bounds of the joint positions and velocities are typically converted into joint-acceleration bounds (Rubrecht et al., 2010)

2.3 Capture Point

Regardless of the task they are performing, legged robots must take care of balancing (i.e. avoiding to fall) at the same time. Balancing is fundamental for legged robots and it has been extensively studied (Collette et al., 2008; Morisawa et al., 2012; Goswami and Kallem, 2004; Hyon and Cheng, 2006; Sherikov et al., 2014). This problem is particularly well understood for robots in contact with a flat terrain only. In this case, the dynamics of the robot CoM c is well approximated by a linear inverted pendulum. In this model the robot is approximated as a point mass (maintained at a constant height) supported by a variable-length leg link (Pratt et al., 2006). The resulting dynamics is:

$$\ddot{c}^{xy}(\phi) = \omega(\phi)^2(c^{xy}(\phi) - u),$$

where $u \in \mathbb{R}^2$ is the Zero Moment Point (ZMP), which is equivalent to the center of pressure (Wieber, 2002), and $\omega(\phi) = \sqrt{\frac{g}{c^z(\phi)}}$. The same dynamics can also be obtained from the real dynamics of the robot CoM, by assuming that $\dot{c}^z = 0$ and the rate of change of the robot angular momentum is null (Wieber et al., 2015). Using this linear dynamics we can compute the point on the ground where the robot can put its ZMP to in order to stop its CoM:

$$\xi(\phi) = c^{xy}(\phi) + \frac{\dot{c}^{xy}(\phi)}{\omega(\phi)}$$

This point is known as the capture point (Pratt et al., 2006), the divergent component of motion or the extrapolated center of mass (Wieber et al., 2015).

2.4 Capture-Point Balance Constraints

Originally, the capture point was used to decide where to make the robot step in order to recover from a push (Pratt et al., 2006). More recently, Ramos et al. (Ramos et al., 2014) proposed use it to ensure the balance of the robot. The key idea is that, as long as the capture point remains inside the convex hull of the contact points (i.e. the so-called *support polygon* $\mathcal{S}$), the robot can balance without taking a step. To ensure the balance of the robot we can then add to (1) another set of inequalities to constrain the capture point to remain inside the support polygon:

$$B(\xi(\phi) + \Delta t \dot{\xi}(\phi)) \leq b,$$

where $\dot{\xi}(\phi) \in \mathbb{R}^2$ is the time derivative of the capture point, and the matrix B and the vector b define the support polygon (i.e. $Bx \leq b \iff x \in \mathcal{S}$). By expressing ξ

and its derivative as functions of c^{xy} and its derivatives we get:

$$B\left(c^{xy}(\phi) + \frac{\dot{c}^{xy}(\phi)}{\omega(\phi)} + \Delta t\left(\dot{c}^{xy}(\phi) + \frac{\ddot{c}^{xy}(\phi)}{\omega(\phi)}\right)\right) \le b$$

$$B\left(c^{xy}(\phi) + \alpha(\phi)\dot{c}^{xy}(\phi) + \frac{\Delta t}{\omega(\phi)}\ddot{c}^{xy}(\phi)\right) \le b,$$

where $\alpha(\phi) = \Delta t + \omega(\phi)^{-1}$. Finally we can express the CoM acceleration $\ddot{c}^{xy}$ as a function of the joint accelerations $\dot{v}$:

$$D(\phi)\dot{v} + B\left(c^{xy}(\phi) + \alpha(\phi)\dot{c}^{xy}(\phi) + \beta(\phi)\right) \le b, \quad (2)$$

where:

$$D(\phi) = \frac{\Delta t}{\omega(\phi)}BJ_{com}(\phi)$$

$$\beta(\phi) = \frac{\Delta t}{\omega(\phi)}\dot{J}_{com}(\phi)v$$

These inequalities are linear with respect to the joint accelerations $\dot{v}$, so they can be added to the QP problem (1) to ensure the robot balance in case of no modeling uncertainties.

3   ROBUSTNESS TO INERTIAL PARAMETER ERRORS

In the previous section we saw that the inertial parameters appear in three different locations in the controller optimization problem (1): i) in the mass matrix M, ii) in the bias forces h, and iii) in the capture-point inequalities (2). Unfortunately M and h depend on ϕ in a highly-nonlinear way, so it is hard to deal with it. In this paper we deal instead with the dependency of the capture-point inequalities (2) on the inertial parameters. More in details, many terms in (2) depend on ϕ, but we will focus on the dependency of the CoM xy coordinates on ϕ. In other words, we want to solve this optimization problem:

$$\underset{y=(\dot{v},f,\tau)}{\text{minimize}} \quad ||Ay - a||^2$$

$$\text{subject to} \quad \begin{bmatrix} M(\hat{\phi}) & -J_c^\top & -S^\top \\ J_c & 0 & 0 \end{bmatrix}\begin{bmatrix} \dot{v} \\ f \\ \tau \end{bmatrix} = \begin{bmatrix} -h(\hat{\phi}) \\ -\dot{J}_c v \end{bmatrix}$$

$$(3a)$$

$$(1a),(1b),(1c)$$

$$D(\hat{\phi})\dot{v} + Bc^{xy}(\phi) \le \bar{b}(\hat{\phi}) \quad \forall \phi \in U, \quad (3b)$$

where $\hat{\phi}$ are the nominal inertial parameters (i.e. those used by the standard controller) and:

$$\bar{b}(\hat{\phi}) = b - B(\alpha(\hat{\phi})\dot{c}^{xy}(\hat{\phi}) + \beta(\hat{\phi}))$$

Problem (3) is not tractable because it has an infinite number of inequality constraints—due to the capture-point inequalities that need to be satisfied for all the possible values of ϕ. In order to solve (3) we need to reformulate it in a tractable form. To do that, we will start by analyzing the relationship between c^{xy} and ϕ (which is linear). Then we will show how to reformulate the robust capture-point inequalities (3b) in a tractable form.

3.1   Dependency of CoM on Inertial Parameters

The CoM of the robot is the average of the CoM of all its links, weighted by their respective masses:

$$c^{xy} = P\frac{\sum_{i=1}^{N}m_i(p_i + {}^wR_i{}^ic_i)}{m_{tot}}$$

$$= \sum_{i=1}^{N}\underbrace{m_{tot}^{-1}P\begin{bmatrix} p_i & {}^wR_i & 0_{3\times 6}\end{bmatrix}}_{F_i}\phi_i \quad (4)$$

$$= \begin{bmatrix} F_1 & \dots & F_N \end{bmatrix}\phi = F\phi,$$

where $P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$, $p_i \in \mathbb{R}^3$ is the position of the reference frame of link i expressed in the world frame, ${}^wR_i \in \mathbb{R}^{3\times 3}$ is a rotation matrix from link i reference frame to the world frame, and m_{tot} is the total mass of the robot. From (4) we can see that the robot CoM is the ratio of two linear functions of the inertial parameters—because m_{tot} is clearly a linear function of ϕ. However, given that we can easily know the robot total mass, we can assume that the uncertainty in m_{tot} be negligible. In the context of robustness we can thus consider c^{xy} as a linear function of ϕ.

3.2   Robust Capture-Point Inequalities

Now we want to reformulate the robust capture-point inequalities into a tractable form. We can start by rewriting the i-th line of (3b) using (4):

$$D_i\dot{v} + B_iF\phi \le \bar{b}_i \qquad \forall \phi \in U,$$

where D_i, B_i and $\bar{b}_i$ are the i-th lines of the associated matrix/vector, and we dropped the dependency on the nominal inertial parameters $\hat{\phi}$ for the sake of readability. We can get rid of the quantifier operator $\forall$ by replacing the uncertain term with its maximum:

$$D_i\dot{v} + \max_{\phi \in U}(B_iF\phi) \le \bar{b}_i \quad (5)$$

We could compute the maximum of $B_iG\phi$ under the constraint of ϕ belonging to the polytope U by solving a Linear Program (LP) for each capture-point inequality. However, that would be too computationally

expensive for a controller that typically has to run at 1 kHz because of the size of the LP (i.e. $10N$ variables and even more constraints). Luckily we show now that we can solve this LP by solving N LPs of much smaller size.

$$\max_{\phi \in U} B_i F \phi = \max_{\phi \in U} \sum_{j=1}^{N} B_i F_j \phi_j = \sum_{j=1}^{N} \max_{\phi_j \in U_j} B_i F_j \phi_j \quad (6)$$

Thanks to this reformulation, rather than maximizing a linear function of the robot CoM, we can maximize a linear function of each link CoM. This boils down to finding, for each link, the CoM position that maximizes the dot product with the vector B_i. If the polytope of possible CoM positions has not many vertices, this optimization can be performed by enumeration. This means that we can compute the dot product of B_i with all the vertices of the CoM polytope and then take the one that resulted in the largest value. Since the vertices of the CoM polytope of each link can be computed offline before starting the controller, this operation is extremely computationally efficient. If we assume that each CoM polytope has n_v vertices and that the support polygon has n_s sides, the computation of $\max_{\phi \in U} B_i F \phi$ for all i requires $n_s n_v N$ dot products of 3D vectors. For a typical scenario of $n_s = 6$, $n_v = 10$ and $N = 30$, this gives 1800 dot products. On a standard computer this would take only a few microseconds, so it is suitable for real-time control on a real robot.

Once this quantity has been computed, the robust capture-point inequalities (3b) can be written as standard linear inequalities and problem (3) can be solved by a standard QP solver.

4 Tests

In this section, we present a series of simulation results that try to answer to the following question: what improvement can we get in terms of fall prevention by using the robust controller? We tested the proposed controller on a typical humanoid tasks (i.e. whole-body reaching) with the 30-degree-of-freedom humanoid robot HRP-2. We carried out several batches of tests, each batch differing for the simulated uncertainties. Each batch was composed by 100 tests, which is not enough for being a statistically significant sampling, but was dictated by the computation time of our simulation environment (about 6 hours for 100 tests). Each test consisted in trying to perform the reaching motion with the two controllers (classic and robust) until the robot either fell or reached the end of the motion. The inertial parameter errors changed at each test, but they were the

Table 1: Simulation parameters.

Symbol	Meaning	Value
Δt	Simulation time step	0.002 s
$\dot{v}_j^{max}$	Max joint acceleration	$10 \, \text{rad} \, \text{s}^{-2}$
v_j^{max}	Max joint velocity	$9.14 \, \text{rad} \, \text{s}^{-1}$
μ	Force friction coefficient	0.3
w_{reach}	Reaching weight	1
w_{post}	Posture weight	10^{-2}
w_{force}	Force minimization weight	10^{-5}

same for the two controllers. We then measured the number of times each controller drove the robot to a fall and the average distance between the final end-effector position and the desired target.
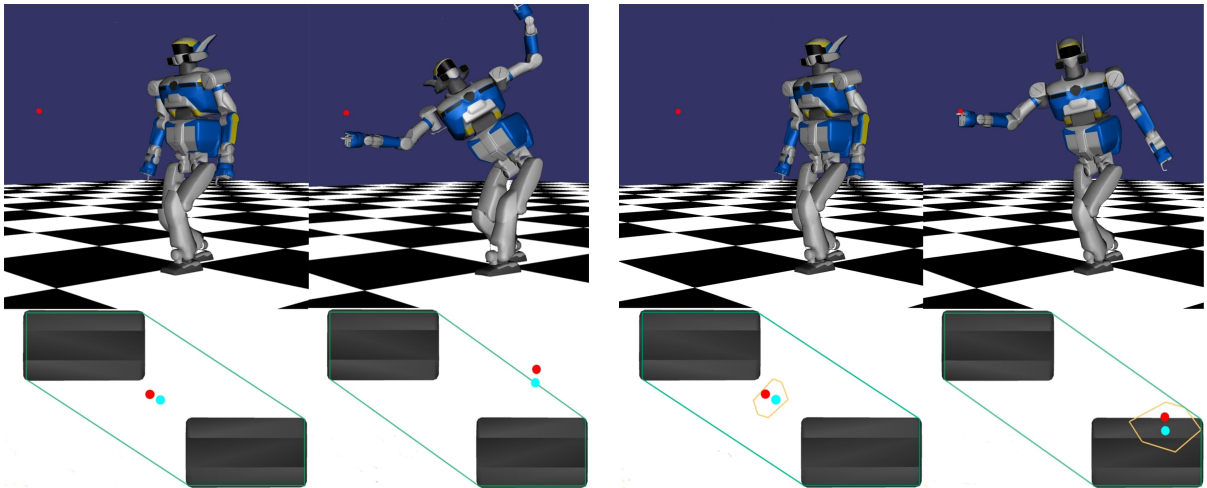
4.1 Simulation Environment

To assess the proposed controller we developed a dedicated simulation environment based on a state-of-the-art algorithm for frictional contacts in multibody systems (Kaufman et al., 2008). We integrated the equations of motion of the system with a first-order Euler scheme with fixed time step Δt. Our choice of not using an off-the-shelf simulator is motivated by our desire to completely understand and control the simulation environment. The inertial parameters (masses and centers of mass) of the model used by the controller did not match those of the model used by the simulator. The random inertial-parameter errors were generated using uniform distribution. For masses, the maximum error was expressed in terms of percentage of the real values. For centers of mass, the maximum error was instead expressed in meters. In each test we specify which uncertainties were simulated. Table 1 lists all the simulation and controller parameters.

4.2 Task Description

The control objective was defined by three tasks that the robot had to perform at the same time. Since the tasks are in conflict, we weighted them with hand-tuned parameters, according to their importance. The three tasks, in order of decreasing priority, are:

- reach the desired target with the right end-effector (weight w_{reach})

- maintain initial joint posture (weight w_{post})

- minimize contact moments and tangential forces (Righetti et al., 2013) (weight w_{force})

We carried out two sets of simulations. In both cases HRP-2 executed a reaching motion that made

(a) Classic control illustrating the robot's loss of balance when the real capture point gets out of the support polygon

(b) Robust control illustrating the robot right end effector reaching close to the goal without losing balance

● Real Capture Point ● Estimated Capture Point ● Support Polygon ● Capture Point Polytope

Figure 1: Screenshots of HRP-2 executing Test 1 to reach the ball target with the robust controller.

its capture point reach the boundaries of its support polygon.

4.3 Test 1

In this test we set the right end-effector target far in front of the robot. Fig. 1 shows some screen shots of the simulations. To reach the target the robot must move its CoM (and hence also its capture point) close to the boundaries of its support polygon. Table 2 presents the results. Regardless of the magnitude of the inertial parameter errors, the robust controller managed to prevent the robot from falling almost always, while with the standard controller the robot fell more than 30% of the times. However, since the target was far away from the robot, the robust controller did not manage to reach it because that would have required violating the robust balance constraints.

4.4 Test 2

In this test we moved the right end-effector target closer to the robot, so that HRP-2 can reach it without moving its CoM close to the support polygon boundaries. However, we increased the desired speed of reaching (by increasing the gains of the reaching task). This affected the velocity of the CoM, which in turns affected the capture point, making it reach the boundaries of the support polygon. The difference with respect to Test 1 is that in this case also the robust controller can reach the target. Table 3 summarizes the results.

Similarly to Test 1, the classic controller leads the robot to a fall in more than 30% of the cases. However, contrary to Test 1, this time the robust controller also manages to reach the target, because it is located closer to the robot. This test shows that being robust does not necessarily implies that we have to sacrifice performance.

5 CONCLUSION

This paper presented a novel optimization-based inverse-dynamics controller that can balance a legged robot despite bounded uncertainties in its inertial parameters. The controller is based on the state-or-the-art control framework Task-Space Inverse Dynamics. In particular, this work is based on the capture-point inequalities (Ramos et al., 2014), which can be included in the controller formulation to ensure the balance of the robot on a level ground. We extended these capture-point inequalities to be robust to bounded uncertainties in the inertial parameters of the robot. The resulting optimization problem is still a Quadratic Program with the same number of variables and inequalities. Moreover, the time required for the additional computation of the robust controller is negligible in this context (i.e. a few microseconds).

We tested the robust controller in simulations with the HRP-2 robot, trying to reach a target position with its right end-effector while balancing. We performed several batches of 100 simulations each, introducing different errors in the inertial parameters and vary-

Table 2: Results of Test 1. For each controller we show the number of falls (Falls), the average time to complete the motion (Task Time) and the average distance of the end-effector to the target at the end of the motion (Task Error).

Uncertainties		Classic Controller			Robust Controller		
Max Mass Error [%]	Max CoM Error[mm]	Falls [%]	Task Time [s]	Task Error [mm]	Falls [%]	Task Time [s]	Task Error [mm]
10	10	31	4.4	49	3	4.5	60
10	20	33	4.3	52	1	4.5	69
10	40	45	4.3	55	3	4.7	102
20	10	38	4.2	49	11	4.5	77
20	20	49	4.5	51	9	4.55	103
20	40	45	4.5	59	14	4.72	122

Table 3: Results of Test 2. For each controller we show the number of falls (Falls), the average time to complete the motion (Task Time) and the average distance of the end-effector to the target at the end of the motion (Task Error).

Uncertainties		Classic Controller			Robust Controller		
Max Mass Error [%]	Max CoM Error[mm]	Falls [%]	Task Time [s]	Task Error [mm]	Falls [%]	Task Time [s]	Task Error [mm]
10	10	29	3.94	2	0	3.4	2
10	20	35	3.4	2	2	3.0	2
10	40	42	3.86	4	0	2.6	4
20	10	43	3.6	3	0	2.8	4
20	20	45	3.5	3	0	2.5	4
20	40	45	3.0	5	0	2.1	5

ing the position of the target position and the required speed of motion. Comparisons against a classic TSID controller have shown impressive improvements in terms of fall prevention.

5.1 Future Work

In the derivation of the robust controller we saw that the inertial parameters appear in different terms of the optimization problem. In this preliminary work we focused only on how the uncertainties affect the CoM position. We believe it should be possible to extend this analysis to the other terms in the capture-point inequalities: CoM velocity, CoM altitude, CoM Jacobian and its time derivative. Extending it also to the mass matrix and the bias forces is an interesting future direction, but it seems more challenging because of nonlinearities.

Another issue of the presented approach is that it is rather conservative. As we saw in Test 1, this can lead to poor performance, which can be unacceptable on a real system. Modeling uncertainties with probability distributions (rather than with polytopes) may lead to a less conservative behavior of the system, and it is thus an interesting future direction. In our previous work (Del Prete and Mansard, 2016) we presented another robust controller, which was robust to joint-torque tracking errors. Integrating the two controllers together seems to be feasible and it would provide robustness to both kinds of uncertainties.

In this preliminary work we focused on simulations to validate the controller formulation and to test it with different parameter errors. Of course, we plan also to test the generated movements on the real HRP-2 robot, to quantify how much it can benefit from this robustness.

REFERENCES

Collette, C., Micaelli, A., Andriot, C., and Lemerle, P. (2008). Robust balance optimization control of humanoid robots with multiple non coplanar grasps and frictional contacts. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3187–3193.

Del Prete, A. and Mansard, N. (2016). Robustness to Joint-Torque Tracking Errors in Task-Space Inverse Dynamics. *IEEE Transaction on Robotics (accepted for publication)*.

Del Prete, A., Nori, F., Metta, G., and Natale, L. (2015). Prioritized Motion-Force Control of Constrained Fully-Actuated Robots: "Task Space Inverse Dynamics". *Robotics and Autonomous Systems*, 63:150–157.

Dietrich, A., Ott, C., and Albu-Schäffer, A. (2013). Multi-objective compliance control of redundant manipulators: Hierarchy, control, and stability. In *2013 IEEE/RSJ international conference on intelligent robots and systems*, pages 3043–3050. IEEE.

Goswami, a. and Kallem, V. (2004). Rate of change of angular momentum and balance maintenance of biped robots. *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, 4.

Herzog, A., Rotella, N., Mason, S., Grimminger, F., Schaal, S., and Righetti, L. (2016). Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid. *Autonomous Robots*, 40(3):473–491.

Hyon, S. and Cheng, G. (2006). Gravity compensation and full-body balancing for humanoid robots. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, volume 00, pages 214–221.

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

Kajita, S. and Kanehiro, F. (2003). Biped walking pattern generation by using preview control of zero-moment point. In *Robotics and Automation (ICRA), 2003 IEEE International Conference on*.

Kaufman, D. M., Sueda, S., James, D. L., and Pai, D. K. (2008). Staggered projections for frictional contact in multibody systems. *ACM Transactions on Graphics*, 27(5):1.

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

Luo, J. and Hauser, K. (2015). Robust Trajectory Optimization Under Frictional Contact with Iterative Learning. In *Robotics, Science and Systems (RSS)*.

Mordatch, I., Lowrey, K., and Todorov, E. (2015). Ensemble-CIO : Full-Body Dynamic Motion Planning that Transfers to Physical Humanoids. In *Robotics and Automation (ICRA), IEEE International Conference on*.

Morisawa, M., Kajita, S., Kanehiro, F., Kaneko, K., Miura, K., and Yokoi, K. (2012). Balance control based on Capture Point error compensation for biped walking on uneven terrain. *IEEE-RAS International Conference on Humanoid Robots*, pages 734–740.

Nguyen, Q. and Sreenath, K. (2015). Optimal Robust Control for Bipedal Robots through Control Lyapunov Function based Quadratic Programs. In *Robotics, Science and Systems (RSS)*.

Pratt, G. (2015). http://www.theroboticschallenge.org/.

Pratt, J., Carff, J., Drakunov, S., and Goswami, A. (2006). Capture Point: A Step toward Humanoid Push Recovery. *2006 6th IEEE-RAS International Conference on Humanoid Robots*.

Pratt, J., Chew, C.-m., Torres, A., Dilworth, P., and Pratt, G. (2001). Virtual model control: An intuitive approach for bipedal locomotion. *The International Journal of Robotics Research*, 20(2):129–143.

Ramdani, N. and Poignet, P. (2005). Robust Dynamic Experimental Identification of Robots With Set Membership Uncertainty. *IEEE/ASME Transactions on Mechatronics*, 10(2):253–256.

Ramos, O. E., Mansard, N., and Soueres, P. (2014). Whole-body Motion Integrating the Capture Point in the Operational Space Inverse Dynamics Control. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*.

Righetti, L., Buchli, J., Mistry, M., Kalakrishnan, M., and Schaal, S. (2013). Optimal distribution of contact forces with inverse dynamics control. *The International Journal of Robotics Research*, (January).

Righetti, L., Buchli, J., Mistry, M., and Schaal, S. (2010). 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. World Scientific Publishing Co. Pte. Ltd.

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

Saab, L., Mansard, N., Keith, F., Fourquet, J.-Y., and Soueres, P. (2011). Generation of dynamic motion for anthropomorphic systems under prioritized equality and inequality constraints. *Robotics and Automation, IEEE International Conference on*, pages 1091–1096.

Saab, L., Ramos, O. E., Mansard, N., Soueres, P., and Fourquet, J.-y. (2013). Dynamic Whole-Body Motion Generation under Rigid Contacts and other Unilateral Constraints. *IEEE Transactions on Robotics*, 29(2):346–362.

Sentis, L. and Khatib, O. (2004). Task-oriented control of humanoid robots through prioritization. *International Journal of Humanoid Robotics*, pages 1–16.

Sherikov, A., Dimitrov, D., and Wieber, P.-b. (2014). Whole body motion controller with long-term balance constraints. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*.

Traversaro, S., Del Prete, A., Ivaldi, S., and Nori, F. (2015). Inertial parameters identification and joint torques estimation with proximal force / torque sensing. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*.

Wieber, P.-B. (2002). On the stability of walking systems. In *Proceedings of the International Workshop on Humanoid and Human Friendly Robotics*, pages 1–7.

Wieber, P.-B., Tedrake, R., and Kuindersma, S. (2015). Modeling and Control of Legged Robots. In *Bruno Siciliano and Oussama Khatib, editors, Springer Handbook of Robotics, 2nd Ed, chapter 48*.