2018 IEEE International Conference on Robotics and Automation (ICRA)
May 21-25, 2018, Brisbane, Australia

Torque-based dynamic walking - A long way from simulation to experiment

Johannes Englsberger, George Mesesan, Alexander Werner, Christian Ott

Abstract— **This paper presents methods that facilitate the implementation of dynamic walking on torque-controlled robots in real world experiments. The work uses the Divergent Component of Motion (DCM) for walking trajectory generation and control. The DCM controller is embedded into a whole-body controller (WBC) that produces a full-body walking behavior. While in simulation the combination of DCM and WBC is sufficient for achieving sophisticated walking gaits, during our initial experiments several real-world issues, detailed in this paper, prevented the original control framework from functioning. This work presents the improvements to the original control framework that enabled a breakthrough on the way to achieving torque-based dynamic walking on a real robot.**

I. Introduction

Legged locomotion is regarded as a difficult problem due to its hybrid dynamics, unilaterality constraints of contact forces and the high dimensionality and nonlinearity of the system dynamics. For online gait generation, a widely used concept is to focus mainly on the robot's center of mass (CoM) dynamics, which covers the most important effects for legged locomotion. This idea was successfully applied to both walking [1]–[4] and running controllers [5]–[7]. In the field of robotic walking control, one of the most popular models is the Linear Inverted Pendulum (LIP) [1], [2], [8]. Recently, the concept of Divergent Component of Motion (DCM) [3], [9], [10] (also known as 'Capture Point' [11]) was introduced to facilitate gait generation and control by only considering the divergent component of the CoM dynamics, while preserving its stable component.

While traditional humanoid robots could only be position-controlled, more recently several torque-controllable robots have been developed [12]–[15]. At its core, locomotion can be regarded as CoM manipulation, which – via Newton's second law $\ddot{\boldsymbol{x}} = \boldsymbol{F}_{com}/m$ – is related to using $\boldsymbol{F}_{com}$ as control input. While position controlled robots produce this required force indirectly, torque-controlled robots provide direct access to $\boldsymbol{F}_{com}$ via joint torque control. And yet, the research community has not yet reached an agreement to the question if position or torque controlled robots are more appropriate for bipedal walking. At this juncture, many successful walking robots are based upon sophisticated mechanical engineering combined with well-tuned control heuristics [16], [17] or they use position controlled joints and admittance control [18]–[20] using feedback from force torque sensors (FTS) in the feet. Position control allows enforcing important walking parameters such as precise foot placement, while admittance control provides sufficient compliance to handle irregular ground surfaces. And yet, torque-controlled robots have several advantages over position-controlled ones: In the context of human-robot interaction, torque-controlled robots are advantageous with regard to clamping safety due to their compliance. Also, torque control provides a higher bandwidth regarding contact force modulation as compared to admittance control, such that more compliant foot-to-ground interaction can be realized. Finally, torque controlled robots allow for interaction with the environment at arbitrary contact points [21], which is advantageous in the context of multi-contact control. Recently, more torque-controlled balancing [15], [22] and walking controllers [4], [23]–[27] have been presented. Most of these controllers are based on the concept of quadratic programming (QP). Feng et al. [24] combine inverse kinematics and inverse dynamics, while Koolen et al. [4] and Hopkins et al. [28] make use of a so called "leaky integrator" (integrating the desired accelerations from the QP to obtain a velocity reference) to overcome the imperfections of the robot hardware. Stephens et al. [23] present push-recovery experiments based on low gain PD joint tracking combined with feedforward torques.

DLR's humanoid robot Toro can be operated both in position and torque control mode. Our position-based walking controller [29] had produced successful dynamic walking gaits on the real hardware. Additionally, we worked on an inverse dynamics based whole-body controller that successfully produced sophisticated walking gaits in simulation [30], [31] and has recently been ported to the actual robotic hardware to achieve *purely torque-based dynamic walking*.

The paper is structured as follows: In Section II, we provide an overview of the robotic platform Toro that was used for simulations and experiments. Section III-A recapitulates our gait generation method. Sections III-B and III-C give an overview of our whole-body control and state estimation frameworks. In section IV, we present simulation results that show the performance of our control framework with regard to smoothness of the reference trajectories and tracking precision. Section V then describes the insights and developed methods that were crucial for the successful porting of the torque-based dynamic walking algorithm to real-world experiments. Section VI presents successful torque-based dynamic walking experiments of the humanoid robot Toro, while section VII concludes the paper.

German Aerospace Center (DLR),
Institute of Robotics and Mechatronics, 82234 Wessling, Germany.
contact: johannes.englsberger@dlr.de

II. Robotic hardware

Both simulations and experiments presented in this paper are performed with the torque-controlled humanoid robot Toro [13], [32]. Toro is a human-size robot (174cm) weighing 76 kg. It has 25 torque-controllable joints: six in each limb and one for vertical torso rotation. Only the two neck DoF cannot be torque-controlled.

Toro is equipped with the following sensors:

- position and torque sensors in torque-controlled joints
- force torque sensor (FTS) above each foot
- inertia measurement unit (IMU) in the trunk
- stereo cameras and other depth sensors in the head

All 25 torque-controlled DoF are used by our control framework, while the neck DoF are omitted. Toro's foot size of 19 cm x 9.5 cm is comparably small regarding its body height, which is a challenge regarding dynamic stability. In contrast to e.g. the Atlas robot [14] or other customized walking machines [16], [17], Toro's drive units – unmodified drive units of the DLR/KUKA light weight robot [33] – are not specifically designed for locomotion tasks. This limits Toro's performance with regard to maximum joint speed and torque.

III. Original control and state estimation framework

While the main contribution of this paper is the presentation of insights and methods that finally allowed to get real-world torque-based walking to work, the following sections will first provide an overview of the basic algorithms – namely our DCM-based walking trajectory generation and tracking, WBC-based whole-body control and state estimation framework – that facilitate torque-based walking in simulation and served as starting point for our efforts towards corresponding experimental evaluations.

A. Walking trajectory generation and control

The concept of DCM stands out due to its simplicity and comprehensibility. This section provides a summary on both its basics and our most recent DCM-related advancements.

1) Review of basics on DCM and VRP: The fundamental theory on DCM $\boldsymbol{\xi}$ and Virtual Repellent Point (VRP) $\boldsymbol{v}$ is found in [3], [30]. The DCM is defined as

$$\boldsymbol{\xi} \;=\; \boldsymbol{x} \;+\; b\,\dot{\boldsymbol{x}} \;. \tag{1}$$

Here, $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ denote the center of mass (CoM) position and velocity, respectively, each being a three-dimensional quantity. The DCM time-constant b can be derived from the average height of the CoM above ground surface Δz_{vrp} as $b = \sqrt{\Delta z_{vrp}/g}$, g being the gravitational constant. Solving (1) for $\dot{\boldsymbol{x}}$, the CoM dynamics are

$$\dot{\boldsymbol{x}} \;=\; -\frac{1}{b}\,(\boldsymbol{x}-\boldsymbol{\xi}) \;, \tag{2}$$

which shows that the CoM follows the DCM with a stable first order dynamics. Thus, for planning we neglect the CoM dynamics, which facilitates the gait design process. Differentiating (1) and with Newton's second law $\ddot{\boldsymbol{x}} = \boldsymbol{F}_{com}/m$ (m being the robot's total mass), we find the following unstable first order dynamics for the DCM:

$$\dot{\boldsymbol{\xi}} \;=\; \frac{1}{b}\,(\boldsymbol{\xi}-\boldsymbol{v}) \;. \tag{3}$$

Here we already inserted the definition of the Virtual Repellent Point (VRP) $\boldsymbol{v}$, which encodes $\boldsymbol{F}_{com}$ via

$$\boldsymbol{F}_{com} \;=\; \frac{m}{b^2}\,(\boldsymbol{x}-\boldsymbol{v}) \;. \tag{4}$$

Looking at (3), we find that the DCM diverges away from the VRP. In [30], [31] this divergent nature of the DCM is used for the design of contact consistent DCM references.

2) Generating consistent multi-step VRP and DCM references: One of the most crucial components of a reliable walking control framework is a walking trajectory generator that produces smooth and feasible CoM (or DCM) trajectories. In [3], we presented a method for continuous double support and heel-to-toe DCM reference generation. Yet, during double support, the method used a third order interpolation of the DCM trajectory that didn't explicitly consider the resulting VRP reference trajectory, which may cause problems regarding contact force feasibility. In [30] and finally in [31], this issue was addressed thoroughly. [31] reformulates the VRP and DCM based trajectory generation problem from [3] in matrix form, such that the DCM reference $\boldsymbol{\xi}_{ref}(t)$ for a given time t is obtained as

$$\boldsymbol{\xi}_{ref}(t) \;=\; \boldsymbol{A}_{\boldsymbol{\xi}_{ref}}(t)\;\boldsymbol{v}_{wp} \;. \tag{5}$$

Here, $\boldsymbol{A}_{\boldsymbol{\xi}_{ref}}(t)$ denotes an analytically derived matrix and $\boldsymbol{v}_{wp}$ is a stacked vector of arbitrary VRP waypoints. The DCM reference $\boldsymbol{\xi}_{ref}(t)$ is smooth and the corresponding VRP reference trajectory $\boldsymbol{v}_{ref}(t)$ continuous (see Fig. 1).

3) DCM tracking control: As in [3], [30], [34], we use a standard DCM tracking controller of the form

$$\boldsymbol{v} \;=\; \boldsymbol{v}_{ref}(t) + (1+k_\xi\, b)\;\underbrace{(\boldsymbol{\xi}-\boldsymbol{\xi}_{ref}(t))}_{\tilde{\boldsymbol{\xi}}} \;. \tag{6}$$

Here, $k_\xi \geq 0$ denotes the DCM tracking gain. This control law uses the nominal VRP reference $\boldsymbol{v}_{ref}(t)$ and the DCM tracking error $\tilde{\boldsymbol{\xi}}$ as input to compute the desired VRP position $\boldsymbol{v}$. Inserting the latter into (4) returns the desired linear CoM force, which is commanded to the whole-body controller.

B. QP-based whole-body control (WBC)

This section provides an overview of our whole-body control (WBC) implementation and quadratic program (QP) setup. Similar to [4], [26], [35], [36], the contact wrenches are represented via polyhedral cones that approximate the friction cones in the corners of each foot. We choose the n_{act} actuated joint torques $\boldsymbol{\tau}_{act}$ and the linear contact force magnitudes $\boldsymbol{\rho}$ (corresponding to the edges of the polyhedral cones) as optimization variables and combine them in a

single vector $\boldsymbol{u} = [\boldsymbol{\tau}_{act}^T, \boldsymbol{\rho}^T]^T$. The general free-floating robot equations of motion in terms of $\boldsymbol{u}$ are given by

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{\tau}_{grav} = \underbrace{\begin{bmatrix}\boldsymbol{S}_{act}, & {}^{\tau}\boldsymbol{A}_{\rho}\end{bmatrix}}_{{}^{\tau}\boldsymbol{A}_u} \underbrace{\begin{bmatrix}\boldsymbol{\tau}_{act} \\ \boldsymbol{\rho}\end{bmatrix}}_{\boldsymbol{u}} , \tag{7}$$

Here, $\boldsymbol{M}$, $\boldsymbol{C}$ and $\boldsymbol{\tau}_{grav}$ denote the inertia and Corioli matrix and the gravitational torque, respectively. The matrices $\boldsymbol{S}_{act}$ and ${}^{\tau}\boldsymbol{A}_{\rho}$ map $\boldsymbol{\tau}_{act}$ and $\boldsymbol{\rho}$ to generalized torques. Equation (7) is often enforced via an equality constraint in the QP formulation. In our work, we decide to pre-solve for the generalized joint accelerations $\ddot{\boldsymbol{q}}$ as

$$\ddot{\boldsymbol{q}} = \underbrace{-\boldsymbol{M}^{-1}\left(\boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{\tau}_{grav}\right)}_{\ddot{\boldsymbol{q}}_{MB}} + \underbrace{\boldsymbol{M}^{-1}\ {}^{\tau}\boldsymbol{A}_u}_{\boldsymbol{Q}}\ \boldsymbol{u} , \tag{8}$$

such that the mentioned equality constraint is no longer required in the QP. All whole-body control tasks are formulated using the control input vector $\boldsymbol{u}$ in the following form:

$$\boldsymbol{d}_{i,des} \stackrel{!}{=} \underbrace{\boldsymbol{D}_i\,\boldsymbol{u}}_{\boldsymbol{d}_i} . \tag{9}$$

Here, $\boldsymbol{d}_{i,des}$ and $\boldsymbol{d}_i$ denote the desired and achieved task vectors of the i-th task, respectively. $\boldsymbol{D}_i$ is the corresponding task mapping matrix. The following table outlines all task matrices and vectors used in this work:

task	task matrix $\boldsymbol{D}_i$	desired task vector $\boldsymbol{d}_{i,des}$
centr. mom.	$[\boldsymbol{0}_{6\times n_{act}}, {}^{com}\boldsymbol{A}_{\rho}]$	$\begin{bmatrix}\boldsymbol{F}_{ext,des} \\ \boldsymbol{\tau}_{com,des}\end{bmatrix}$
foot tracking	$\boldsymbol{J}_f\,\boldsymbol{Q}$	$\begin{bmatrix}\ddot{\boldsymbol{x}}_{f,des} \\ \dot{\boldsymbol{\omega}}_{f,des}\end{bmatrix} - \dot{\boldsymbol{J}}_f\,\dot{\boldsymbol{q}} - \boldsymbol{J}_f\,\ddot{\boldsymbol{q}}_{MB}$
torso orient.	$\boldsymbol{J}_{\omega,torso}\,\boldsymbol{Q}$	$\dot{\boldsymbol{\omega}}_{torso,des} - \dot{\boldsymbol{J}}_{torso}\,\dot{\boldsymbol{q}} - \boldsymbol{J}_{torso}\,\ddot{\boldsymbol{q}}_{MB}$
posture	$\boldsymbol{S}_{post}\,\boldsymbol{Q}\,\boldsymbol{u}$	$\ddot{\boldsymbol{q}}_{post,des} - \boldsymbol{S}_{post}\,\ddot{\boldsymbol{q}}_{MB}$
regulariz.	$\boldsymbol{I}_{n_u\times n_u}$	$\boldsymbol{0}$

Here, ${}^{com}\boldsymbol{A}_{\rho}$ maps $\boldsymbol{\rho}$ to the corresponding centroidal wrench, the Jacobian $\boldsymbol{J}_f$ corresponds to the 6D foot motion, the Jacobian $\boldsymbol{J}_{\omega,torso}$ corresponds to the torso's angular velocity and $\boldsymbol{S}_{post}$ selects the joints for the postural task. The desired external force $\boldsymbol{F}_{ext,des}$ and torque $\boldsymbol{\tau}_{com,des}$ are provided by the DCM-based walking controller (4) and (6) and an angular momentum regularization controller. The desired acceleration quantities, i.e. $\ddot{\boldsymbol{x}}_{f,des}$, $\dot{\boldsymbol{\omega}}_{f,des}$, $\dot{\boldsymbol{\omega}}_{torso,des}$ and $\ddot{q}_{post,des}$, are computed via task space PD controllers. For further details on our task space PD controller formulation the reader is referred to [30]. The last row in the table above denotes a regularization task that tries to minimize the optimization variables. Reordering (9), we express the task errors as

$$\Delta\boldsymbol{d}_i = \boldsymbol{d}_i - \boldsymbol{d}_{i,des} = \boldsymbol{D}_i\,\boldsymbol{u} - \boldsymbol{d}_{i,des} . \tag{10}$$

where the index i stands for all task indices "centroidal momentum", "left foot", "right foot", "torso","posture" and "regularization". Now, the task errors are used to write a cost function G, which is minimized via the following QP:

$$\begin{aligned} \min_{\boldsymbol{u}} \quad & G = \frac{1}{2}\ \sum_i (\Delta\boldsymbol{d}_i^T\ \boldsymbol{W}_i\ \Delta\boldsymbol{d}_i) \\ \text{s.t.} \quad & \underline{\boldsymbol{\tau}}_{act} \le \boldsymbol{\tau}_{act} \le \overline{\boldsymbol{\tau}}_{act} \\ & \boldsymbol{0} \le \boldsymbol{\rho} \end{aligned} . \tag{11}$$

We choose diagonal task weighting matrices $\boldsymbol{W}_i$ (one for each task i) to independently weight the different task elements. The inequality constraints guarantee that the actuated joint torques $\boldsymbol{\tau}_{act}$ remain within specified bounds and the linear contact force magnitudes $\boldsymbol{\rho}$ are non-negative, such that the resulting contact wrenches are feasible.

C. State estimation

While in simulation environments the robot state is known, on a real robot platform it has to be estimated (if no external tracking system is available). The state estimation framework used in this work originated from the one presented by Henze et al. [22], who – in contrast to e.g. [38], [39] that use Kalman filters – aims at a direct and state-less estimation of base frame coordinates and velocities. It fuses contact information, kinematics and inertial measurements.

1) Fusion of base frame transformation: In the presented work, the transformation of the robot's free-floating base is computed by fusing the base link rotation, as provided by the IMU, with foot odometry information. In our original setup, support phase changes are triggered by the time-based walking state machine. When a support change trigger is activated, the current estimated foot location (or the center point in between the feet in case of double support) is stored. This stored location is used as "anchor" position, i.e. after the anchoring, the corresponding "anchored" position is assumed to be fixed, until the next contact switch. The foot orientation is neglected which achieves higher robustness of the state estimation when walking on compliant ground surfaces with low rotational contact stiffness. The foot anchor constrains three degrees of freedom (DOF) out of the six DOF that describe the free-floating base transformation. The three missing DOF – namely the base link orientation – are provided directly by the IMU orientation measurement/estimate. Note: slipping foot contacts may cause a drift in the state estimate due to the absence of absolute positioning sensors.

2) Fusion of base velocities: The floating base velocity is estimated based on the following measurement equation:

$$\underbrace{\begin{bmatrix}\dot{\boldsymbol{x}}_{lf} \\ \dot{\boldsymbol{x}}_{rf} \\ {}_b\boldsymbol{\omega}_{IMU} \\ \dot{\boldsymbol{q}}_{joint,msr} \\ \dot{\boldsymbol{x}}_{base,est}\end{bmatrix}}_{\boldsymbol{c}_v} = \underbrace{\begin{bmatrix}{}_h\boldsymbol{J}_{\dot{\boldsymbol{x}}_{lf}} \\ {}_h\boldsymbol{J}_{\dot{\boldsymbol{x}}_{rf}} \\ {}_b\boldsymbol{J}_{\boldsymbol{\omega}_{IMU}} \\ [\boldsymbol{0}_{n_{act}\times 6}, \boldsymbol{I}_{n_{act}\times n_{act}}] \\ {}_h\boldsymbol{J}_{\dot{\boldsymbol{x}}_{base}}\end{bmatrix}}_{\boldsymbol{C}_v} \dot{\boldsymbol{q}} . \tag{12}$$

Here, the velocity constraint matrix $\boldsymbol{C}_v$ maps the generalized velocities $\dot{\boldsymbol{q}}$ to the velocity constraint vector $\boldsymbol{c}_v$. The hybrid

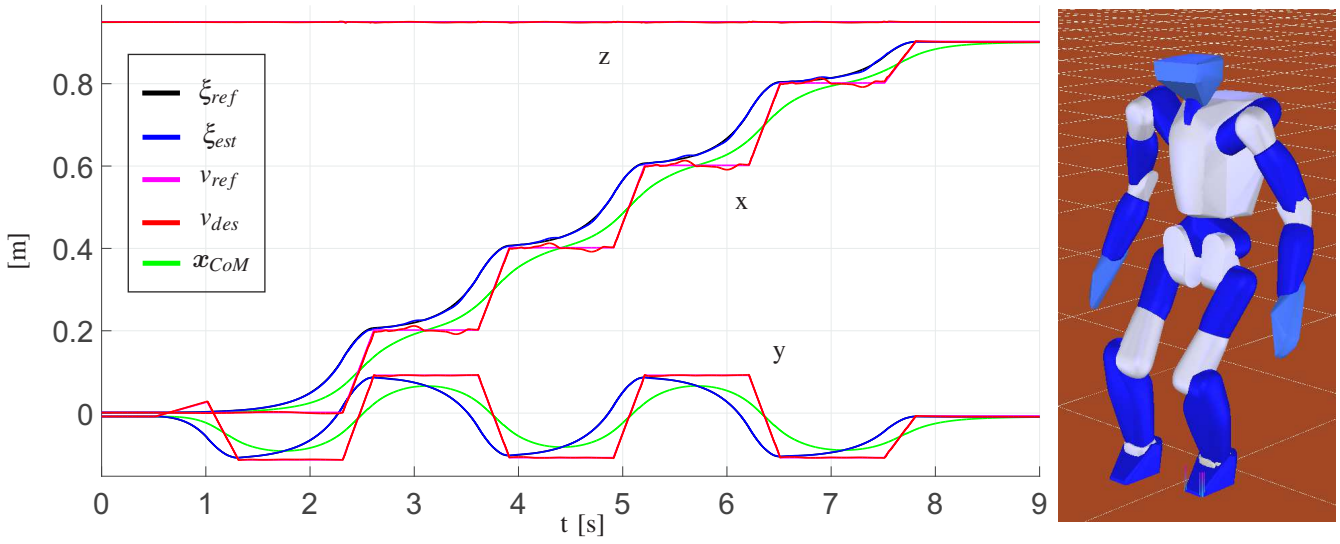

Fig. 1. Simulation of humanoid robot Toro [13] walking in OpenHRP [37]. Step length: 20 cm, single/double support times: 1 sec. / 0.3 sec.

Jacobians ${}_h\boldsymbol{J}_{\dot{\boldsymbol{x}}_{lf}}$ and ${}_h\boldsymbol{J}_{\dot{\boldsymbol{x}}_{rf}}$ correspond to the left and right linear foot velocities $\dot{\boldsymbol{x}}_{lf} \in \mathbb{R}^3$ and $\dot{\boldsymbol{x}}_{rf} \in \mathbb{R}^3$, which cannot be measured but are based on *assumptions* (during stance they are assumed to be zero). The IMU body Jacobian (see [30], [40]) ${}_b\boldsymbol{J}_{\omega_{IMU}}$ maps the generalized velocities to measured gyro rates ${}_b\boldsymbol{\omega}_{IMU}$. The fourth row in (12) encodes the equation for the joint velocity measurement (obtained via numerical differentiation of the joint position measurements). The last row in (12) is introduced to mitigate the sensitivity of the linear base velocity estimate w.r.t. gyro noise. The linear base velocity estimate $\dot{\boldsymbol{x}}_{base,est}$ is obtained by integrating the IMU acceleration (corrected by the gravity vector) and adding it to the previous linear base velocity estimate. Note that equation (12) is over-determined. We solve for the generalized joint velocities $\dot{\boldsymbol{q}}$ via weighted pseudo-inverse

$$\dot{\boldsymbol{q}} = (\boldsymbol{C}_v^T\, \boldsymbol{W}(\boldsymbol{t})\, \boldsymbol{C}_v + \Lambda)^{-1}\, \boldsymbol{C}_v^T\, \boldsymbol{W}(\boldsymbol{t})\, \boldsymbol{c}_v \ . \qquad (13)$$

Here, Λ denotes a regularization term and $\boldsymbol{W}(\boldsymbol{t})$ is a diagonal matrix used to weight the different measurements[1]. The weights corresponding to the linear foot velocity measurements are of particular importance. At the end of stance, they are ramped down to zero (and kept there during foot swing), while they are ramped up at the beginning of a new stance phase. While equation (13) solves for the whole vector of generalized velocities, in experiments we encountered improved robot performance if only the base velocities are selected from $\dot{\boldsymbol{q}}$ in (13), whereas the actuated joint velocity estimates are chosen to equal the measured joint velocities $\dot{\boldsymbol{q}}_{joint,msr}$ (derived via differentiation of joint positions).

[1]Note: as compared to the covariance matrices in Kalman filters, these weights are more intuitive to tune.

IV. Simulation Results

In the previous sections we presented our algorithms for DCM-based walking trajectory generation and tracking, QP-based whole-body control and state estimation. Fig. 1 shows the result of a walking simulation performed in the OpenHRP simulator [37], in which the humanoid robot Toro [13] initially stands, smoothly transitions to walking (see caption for gait parameters) and back to standing. The purpose of showing this simulation result is twofold: firstly, it shows the smooth/continuous – even across different walking states (standing - walking - standing) – DCM reference trajectories (black) and corresponding VRP reference trajectories. Secondly, the simulation shows the remarkable tracking performance (note that the DCM reference trajectory (black) is hardly visible below the estimated DCM trajectory ξ_{est} (blue)) of the overall control and estimations framework as presented in the previous sections. Note: also the methods described in Sec. V are already activated here and have no adverse effects on the robot control performance. A multitude of further OpenHRP simulations has been performed including faster and longer stepping, sensor noise and step adjustment (see [31]), which all demonstrate convincing performance of the proposed control and estimation framework.

V. Towards Real-World Torque-Based Dynamic Walking

In the previous section we showed that our proposed control and estimation framework is suitable for achieving torque-based dynamic walking in simulation. Unfortunately, in the real world a robot is subject to imperfections including:

- sensor noise, offsets and drift
- communication delays
- modelling errors including robot inertias and unmodeled joint friction, link and joint elasticities

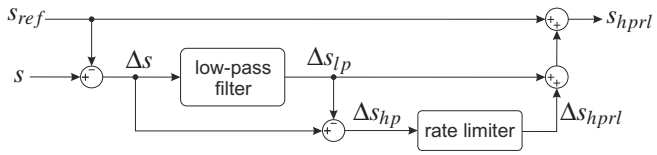

Fig. 2. Overview on the concept of High-Pass Rate Limiter (HPRL)

- non-perfect torque source, e.g. regarding bandwidth[2]

This section describes the methods that contributed to the successful implementation of torque-based walking on Toro.

A. High-Pass Rate Limiter

When the basic control and state estimation framework as described above was first run on Toro, high amplitude oscillations were encountered, even for the case of standing. We want to point out that during the first trials, no sensor filtering was activated and we tried to achieve the CoM damping that naturally (see [30] for details) correlates to our preferred DCM gain $k_\xi = 4$ (as used in (6); corresponding to a proportional CoM gain of 984 N/m and a damping gain of 552 Ns/m). Only after a reduction of the CoM damping by using a damping ratio of 0.2 and additionally applying low-pass filters, we succeeded in suppressing the previously observed oscillations for the most part, such that stable torque-based standing/balancing was achieved. In contrast to balancing control (see [22]), where low damping ratios may be acceptable, walking control requires high effective damping ratios in the range of 1. Note: due to natural friction in the robot's joints, the nominal damping ratio can be reduced to some part, depending on the robot hardware. On Toro however, due to low friction, overshooting behavior could be observed up to a nominal damping ratio of 0.85. Even when trying to walk in simulation, we found that for low damping ratios (corresponding to complex eigenvalues) the robot's CoM tends to overshoot, which causes the required control inputs to exceed the constraints (desired CoP leaves the base of support) and causes the robot to fall immediately. Our goal was thus to first gradually increase the CoM damping back to the desirable level, before trying to actually walk. When trying to increase the CoM damping in experiment, we observed that this could only be done (without provoking increased oscillations) by lowering the eigenvalues of the previously mentioned first order filters. This meant that we could achieve high nominal damping, but on the other side the phase lag introduced by the first order filters now once again caused the robot's CoM to overshoot. This motivated our search for lag-free filtering solutions.

One possible solution are *rate limiters*. As long as the maximum allowable rates are not exceeded, rate limiters are *completely inactive*, while becoming active only during strong noise or vibrations. For tracking applications, the direct usage of rate limiters comes with a drawback: the measured signal is the sum of undesired noise/vibrations and the actual signal that typically follows the reference signal. Thus, standard rate limiters cause an asymmetric cut down behavior with respect to the undesired noise/vibrations and thus lag effects similar to low-pass filters (see Fig. 3). For high actual/reference signal rates, the rising slope of the noise/oscillation signal is cut down more than its falling slope, which is undesirable. Even worse, for high actual/reference rates a standard rate limiter will even start to cut down these which heavily corrupts the resulting signal.

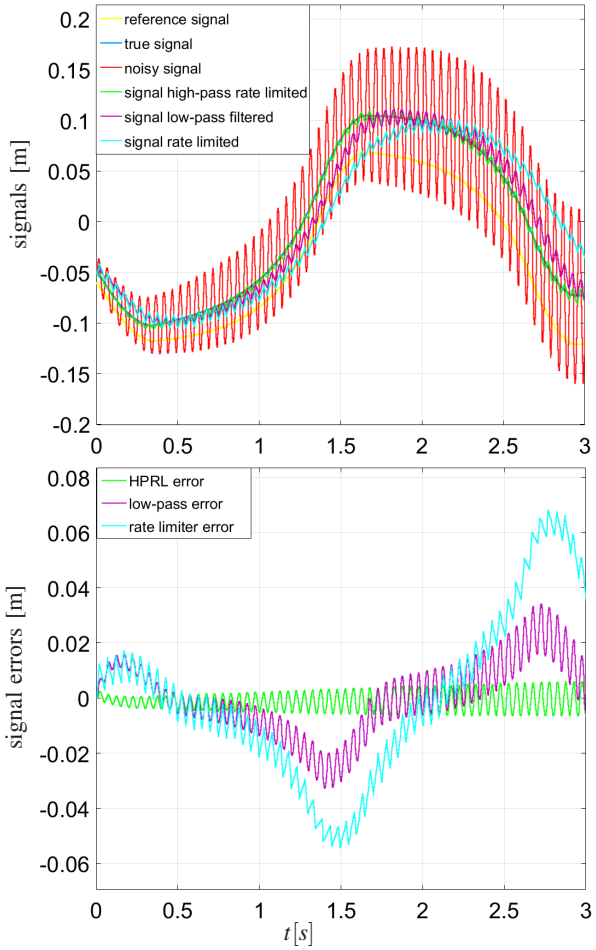

Fig. 3. Comparison of filter performances of High-Pass Rate Limiter (HPRL, green), low-pass filter (violet) and standard rate limiter (cyan) applied to an oscillatory DCM signal. The HPRL causes insignificant lag while successfully reducing the noise/vibration level.

Our solution to this issue is a High-Pass Rate Limiter (HPRL, see Fig. 2). We first subtract the reference signal s_{ref} from the original signal s to obtain the delta signal Δs. Now we use a low-pass filter (implemented as first order filter) to split the delta signal into a slower (low-pass, approximating the actual delta signal) component Δs_{lp} and a faster (high-pass, approximating noise and undesired high-

[2]Note: we believe that the dominant imperfections on Toro are unmodeled link and joint elasticities and its limited torque bandwidth, while joint velocities derived via discrete differentiation of joint positions are unproblematic.

frequency oscillations) component Δs_{hp}. The rate limiter is then only applied to the high-pass signal, its result Δs_{hprl} being added back to the reference and low-pass signals to yield the output signal s_{hprl}. This way the rate limiter can (where required) be adjusted to very low cut off rates, while the described problem of asymmetric signal cutting is avoided to the most part. In the most extreme case, where the allowable rate is set to zero, the HPRL results in a low-pass filter behavior. By careful tuning, the HPRL is adjusted to cause insignificant lag (see figures 1 and 3), while drastically reducing noise and high-frequency oscillations of the signal, which would otherwise be passed into the controller and thus (re-)enter the overall closed control loop.

By careful analysis of our system (including state estimation, controllers and the actual robot), we found that any filtering (wherever possible) should take place *not directly after the measurements* but instead right before the controllers, i.e. in task space. Firstly, this avoids a distortion of the filtering operation via time-varying transformations of the signal[3] and secondly (most important for our HPRL concept) only at that level the corresponding reference velocity signals are available. In our implementation, we apply HPRLs to *every signal that is part of the feedback loop* (remember: the HPRLs are completely *inactive* in case of low noise/vibration levels). While the most critical signal is the DCM measurement, the filtering of all the other signals mainly helps taming the vibrations once they are already in the system. After the introduction of HPRL filtering, the desired CoM damping gain (damping ratio = 1) could finally be achieved[4] and this without the CoM overshooting behavior that was observed when using low-pass filters. In addition to the HPRLs we limit the maximum rates of our optimization variables via variable bounds (see inequalities in (11)).

B. Improved contact reliability and state estimation

The foot odometry and velocity estimate as described in sections III-C.1 and III-C.2 are based on a contact reliability estimate (compare to [41], [42]). Initially, the latter was purely based on our (time-based) walking state machine. While jumps in the contact reliability estimate were avoided via smooth transitions, the purely time-based estimate without feedback about the ground reaction forces (GRF) was sensitive w.r.t. foot tracking errors: during foot touch down the foot was sometimes assumed to have zero velocity (corresponding to raised weights in the pseudo-inverse (13)) while actually still moving. Similarly, when the nominal walking state switched, the foot anchor as described in Sec. III-C.1 was set, which corrupted the foot odometry.

To improve the robustness of the state estimation, the contact reliability estimate of each foot was augmented by force torque measurements (from the FTS). The force/torque-based reliability estimate ramped up from 0 to 1 between two thresholds. In the final setup, these thresholds are set to 26% and 33% of Toro's weight. The final contact reliability was then chosen as the minimum of time-based and GRF-based contact reliability estimates and the foot velocity weights in (13) were scaled accordingly. Similarly, the trigger for anchoring the current foot position estimate was only initiated once the contact force raised above a certain level (here: 31% of Toro's weight). This combined time- and contact force based foot odometry and foot velocity estimate considerably improved our walking performance.

C. Robustification of swing foot touch-down

In our original implementation, the foot acceleration task switched from PD-based tracking to a *zero acceleration assumption* (i.e. $\ddot{\boldsymbol{x}}_{foot} = \mathbf{0}$) at the *nominal transition* from foot swing to stance. In case of good foot tracking (as typically observed in simulations) this approach works sufficiently well. If in contrast the controller switches to the $\ddot{\boldsymbol{x}}_{foot} = \mathbf{0}$ assumption while still in the air due to foot tracking errors (as observed during our experiments), the touchdown of the foot becomes shaky and the whole swing leg may diverge from its intended posture. In this case, the commanded foot wrench that typically points towards the ground is the only effect that pushes the foot towards the floor, while neither foot velocity, nor position are contained (as e.g. by a PD controller). To increase the robustness of the foot settling, we decided to keep the PD term of the foot tracking controller fully active until a certain threshold in the contact force is reached (similar to the contact reliability estimation described above). The PD terms are then gradually faded out until a second threshold is reached. In the presented implementation, these threshold values were set to 7% and 20% of Toro's weight. This procedure greatly improved the foot positioning precision and reduced the corresponding vibrations at the foot settling process.

Note: In case of remaining foot tracking errors w.r.t. the originally desired foot trajectories, keeping the PD terms active at all times (i.e. throughout stance) would disturb the control performance due to wrong assumed foot accelerations, which requires to turn the PD terms off at some point.

VI. Experimental evaluation

As mentioned above, the basic control and state estimation framework as described in Sec. III had failed to produce satisfactory experimental results. The robustification methods as described in Sec. V were developed in a laborious process that involved a large number of experiments and cross-validation simulations. The first considerable improvement was achieved by introducing the proposed High-Pass Rate Limiters (see Sec. V-A) that finally rendered the required (for walking) comparably high CoM damping possible while allowing for safe well-damped balancing of Toro. An experiment during which Toro balanced on a pile of gym

[3]Example: filtering joint velocities via first order filters and transforming them via a time-varying Jacobian distorts the resulting task space velocity, while filtering the velocity only in task space results in a clean phase lag.

[4]We could even increase the damping ratio up to 1.6 (overdamped), which we used to provoke vibrations and improve the HPRL tuning.

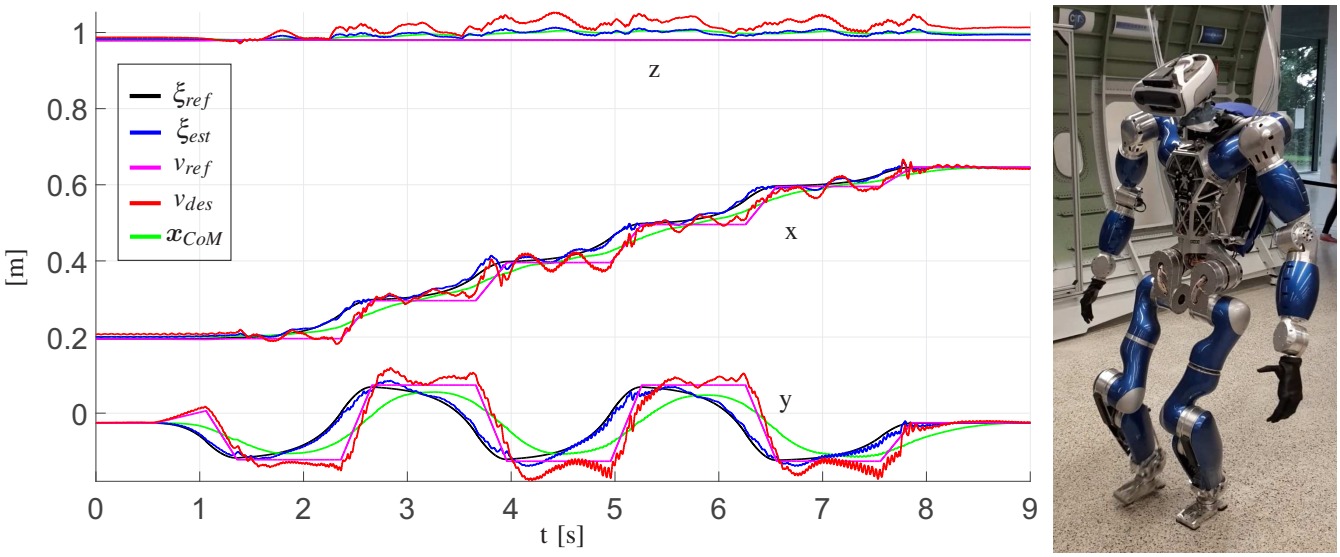

Fig. 4. Torque-based dynamic walking experiment performed by Toro. Step length: 10 cm, single/double support times: 1 sec. / 0.3 sec.

mats validated the performance of our control and estimation framework w.r.t. stationary standing on compliant grounds. The problems related to corrupted foot odometry and velocity estimation and the shaky foot settling were eliminated by the methods as described in sections V-B and V-C. Figure 4 presents the results of an experiment during which Toro successfully performs *purely torque-based dynamic walking*. The DCM trajectories (black and blue) are tracked well. The resulting desired VRP trajectories (red) are close to the VRP references (magenta) and comparably smooth. Note the slight asymmetry in the tracking performance and corresponding VRP deviations when comparing left and right support phases, which shows the imperfections of the real hardware that the controller has to cope with. In addition to the presented experiment, we have achieved successful backward walking and a maximum forward step length of 12cm (for single/double support times of 1s/0.3s).

The proposed torque-based whole-body controller has a considerably increased number of feedback channels as compared to our position control based walking algorithm [29], which increases the chances of exciting natural eigenmodes of the robot due to unmodeled structural elasticities. This, together with the to date suboptimal foot trajectory tracking, explains the current lower maximum step length of 12cm for our torque-controlled walking as compared to the approximately 25cm achieved in position control mode.

VII. Conclusion and Future Work

In this paper, we first reviewed our methods for DCM-based walking trajectory generation, QP-based whole-body control and state estimation. This overall framework was shown to produce sophisticated torque-based dynamic walking gaits in simulation that stand out due to their smoothness and good tracking performance. We then presented several improvements to the original control and state estimation framework that rendered the experimental implementation of torque-based dynamic walking on our humanoid robot Toro possible. These improvements include a High-Pass Rate Limiter (HPRL) that drastically reduced the observed vibrations without adding phase lag (in contrast to first order filters), which allowed for the implementation of the CoM damping levels required for walking, and methods for improved robustness of state estimation and foot touch-down.

In our future research, we will implement passivity-based dynamic walking based on [22] and [43] and compare it to the inverse dynamics based walking controller presented in this paper. For further evaluation of our state estimation and control frameworks we plan to integrate an external tracking system, both for offline verification and online control. Additionally, we intend to improve the walking performance to increase step length and walking speed, and to experimentally verify our DCM-based step adjustment method [31]. Finally, we will validate the performance of our framework on compliant and unstructured ground surfaces.

Acknowledgements

This work is embedded into the COMANOID project [44], which is supported by the European Commission (H2020-ICT-645097 COMANOID).

References

[1] P.-B. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2006, pp. 137–142.

[2] R. Tedrake, S. Kuindersma, R. Deits, and K. Miura, "A closed-form solution for real-time zmp gait generation and feedback stabilization," in *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, 2015, pp. 936–940.

[3] J. Englsberger, C. Ott, and A. Albu-Schäffer, "Three-dimensional bipedal walking control based on divergent component of motion," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 355–368, 2015.

[4] T. Koolen, S. Bertrand, G. Thomas, T. De Boer, T. Wu, J. Smith, J. Englsberger, and J. Pratt, "Design of a momentum-based control framework and application to the humanoid robot atlas," *International Journal of Humanoid Robotics*, vol. 13, no. 01, p. 1650007, 2016.

[5] P. M. Wensing and D. E. Orin, "High-speed humanoid running through control with a 3d-slip model," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 5134–5140.

[6] J. Englsberger, P. Kozlowski, and C. Ott, "Biologically inspired deadbeat controller for bipedal running in 3d," in *IEEE/RSJ Int. Conf. on Intell. Robots and Systems*, 2015, pp. 989–996.

[7] J. Englsberger, P. Kozlowski, C. Ott, and A. Albu-Schäffer, "Biologically inspired deadbeat control for running: from human analysis to humanoid control and back," *Robotics, IEEE Transactions on*, 2016.

[8] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation," in *IEEE Int. Conf. on Robotics and Automation*, 2001, pp. 239–246.

[9] T. Takenaka, T. Matsumoto, and T. Yoshiike, "Real time motion generation and control for biped robot -1st report: Walking gait pattern generation-," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1084–1091.

[10] M. A. Hopkins, D. W. Hong, and A. Leonessa, "Humanoid locomotion on uneven terrain using the time-varying divergent component of motion," in *2014 IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 266–272.

[11] T. Koolen, T. D. Boer, J. Rebula, A. Goswami, and J. E. Pratt, "Capturability-based analysis and control of legged locomotion. part 1: Theory and application to three simple gait models," *Int. J. of Robotics Research*, vol. 31, no. 9, pp. 1094–1113, 2012.

[12] G. Cheng, S.-H. Hyon, J. Morimoto, A. Ude, G. Colvin, W. Scroggin, and S. C. Jacobsen, "CB: A humanoid research platform for exploring neuroscience," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2006.

[13] J. Englsberger, A. Werner, C. Ott, B. Henze, M. A. Roa, G. Garofalo, R. Burger, A. Beyer, O. Eiberger, K. Schmid, and A. Albu-Schäffer, "Overview of the torque-controlled humanoid robot toro," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2014, pp. 916–923.

[14] "Atlas - the agile anthropomorphic robot." [Online]. Available: https://www.bostondynamics.com/atlas0

[15] S. H. Hyon, D. Suewaka, Y. Torii, and N. Oku, "Design and experimental evaluation of a fast torque-controlled hydraulic humanoid robot," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 2, pp. 623–634, 2017.

[16] C. Hubicki, J. Grimes, M. Jones, D. Renjewski, A. Spröwitz, A. Abate, and J. Hurst, "Atrias: Design and validation of a tether-free 3d-capable spring-mass bipedal robot," *The International Journal of Robotics Research*, vol. 35, no. 12, pp. 1497–1521, 2016.

[17] "Robots, designed to be agile and efficient – cassie." [Online]. Available: http://www.agilityrobotics.com/robots/

[18] J. Urata, K. Nshiwaki, Y. Nakanishi, K. Okada, S. Kagami, and M. Inaba, "Online walking pattern generation for push recovery and minimum delay to commanded change of direction and speed," in *Int. Conf. on Intell. Robots and Systems*, 2012.

[19] R. Wittmann, A.-C. Hildebrandt, D. Wahrmann, F. Sygulla, D. Rixen, and T. Buschmann, "Model-based predictive bipedal walking stabilization," in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*, 2016, pp. 718–724.

[20] "Asimo - the world's most advanced humanoid robot." [Online]. Available: http://asimo.honda.com/

[21] B. Henze, A. Dietrich, M. A. Roa, and C. Ott, "Multi-contact balancing of humanoid robots in confined spaces: Utilizing knee contacts," in *Int. Conf. on Intell. Robots and Systems (accepted)*, 2017.

[22] B. Henze, M. A. Roa, and C. Ott, "Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios," *The International Journal of Robotics Research*, vol. 35, no. 12, pp. 1522–1543, 2016.

[23] B. Stephens and C. Atkeson, "Push recovery by stepping for humanoid robots with force controlled joints," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2010, pp. 52–59.

[24] S. Feng, X. Xinjilefu, C. G. Atkeson, and J. Kim, "Optimization based controller design and implementation for the atlas robot in the darpa robotics challenge finals," in *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, 2015, pp. 1028–1035.

[25] M. A. Hopkins, D. W. Hong, and A. Leonessa, "Compliant locomotion using whole-body control and divergent component of motion tracking," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 5726–5733.

[26] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, vol. 40, no. 3, pp. 429–455, 2016.

[27] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Autonomous Robots*, vol. 40, no. 3, pp. 473–491, 2016.

[28] M. A. Hopkins, D. W. Hong, and A. Leonessa, "Compliant locomotion using whole-body control and divergent component of motion tracking," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[29] J. Englsberger, C. Ott, M. A. Roa, A. Albu-Schäffer, and G. Hirzinger, "Bipedal walking control based on capture point dynamics," in *Int. Conf. on Intell. Robots and Systems*, 2011, pp. 4420–4427.

[30] J. Englsberger, "Combining reduced dynamics models and whole-body control for agile humanoid locomotion," Dissertation, Technische Universität München, München, 2016.

[31] J. Englsberger, G. Mesesan, and C. Ott, "Smooth trajectory generation and push-recovery based on divergent component of motion," in *IEEE/RSJ Int. Conf. on Intell. Robots and Systems (accepted)*, 2017.

[32] B. Henze, A. Werner, M. A. Roa, G. Garofalo, J. Englsberger, and C. Ott, "Control applications of TORO - a torque controlled humanoid robot (video)," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2014.

[33] G. Hirzinger, A. Albu-Schäffer, M. Hähnle, I. Schaefer, and N. Sporer, "On a new generation of torque controlled light-weight robots." in *IEEE Int. Conf. on Robotics and Automation*, 2001, pp. 3356–3363.

[34] J. Englsberger, C. Ott, and A. Albu-Schäffer, "Three-dimensional bipedal walking control using divergent component of motion," in *Int. Conf. on Intell. Robots and Systems*, 2013, pp. 2600–2607.

[35] L. Righetti, J. Buchli, M. Mistry, M. Kalakrishnan, and S. Stefan, "Optimal distribution of contact forces with inverse dynamics control," *International Journal of Robotics Research*, vol. 32, no. 3, pp. 280–298, 2013.

[36] P. M. Wensing and D. E. Orin, "Generation of dynamic humanoid behaviors through task-space control with conic optimization," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013, pp. 3103–3109.

[37] F. Kanehiro, K. Fujiwara, S. Kajita, K. Yokoi, K. Kaneko, H. Hirukawa, Y. Nakamura, and K. Yamane, "Open architecture humanoid robotics platform," in *IEEE Int. Conf. on Robotics and Automation*, 2002, pp. 24–30 vol.1.

[38] M. Bloesch, C. Gehring, P. Fankhauser, M. Hutter, M. A. Hoepflinger, and R. Siegwart, "State estimation for legged robots on unstable and slippery terrain," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, 2013, pp. 6058–6064.

[39] X. Xinjilefu, S. Feng, W. Huang, and C. G. Atkeson, "Decoupled state estimation for humanoids using full-body dynamics," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 195–201.

[40] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[41] M. Camurri, M. Fallon, S. Bazeille, A. Radulescu, V. Barasuol, D. G. Caldwell, and C. Semini, "Probabilistic contact estimation and impact detection for state estimation of quadruped robots," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1023–1030, 2017.

[42] J. Hwangbo, C. D. Bellicoso, P. Fankhauser, and M. Huttery, "Probabilistic foot contact estimation by fusing information from dynamics and differential/forward kinematics," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 3872–3878.

[43] G. Mesesan, J. Englsberger, B. Henze, and C. Ott, "Dynamic multi-contact transitions for humanoid robots using divergent component of motion," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, 2017, pp. 4108–4115.

[44] "Comanoid project - multi-contact collaborative humanoids in aircraft manufacturing." [Online]. Available: http://www.comanoid.cnrs.fr/