Dynamic Multi-contact Transitions for Humanoid Robots using Divergent Component of Motion

George Mesesan, Johannes Englsberger, Bernd Henze, Christian Ott

Abstract— **This paper presents a new method for planning and controlling dynamic multi-contact motions for humanoid robots. Our motion planner takes a sequence of multi-contact stances and generates closed-form reference trajectories for the robot center of mass (CoM) position, velocity, and acceleration, based on the concept of Divergent Component of Motion (DCM). The timing of the contact transitions and the end-effector trajectories are automatically computed such that the motion is feasible with respect to kinematic and dynamic constraints. We verify the constraints using a simplified model of the robot to achieve a very fast planner that finds a feasible solution within a few seconds. The reference trajectories serve as inputs to a passivity-based whole-body controller which includes a DCM controller for tracking the CoM trajectory. We demonstrate the robustness of our approach in simulation and experiments with the humanoid robot TORO.**

I. INTRODUCTION

We envision humanoid robots performing tasks that are repetitive, strenuous or too dangerous for humans. Their elevated center of mass and small base of support make dynamic motion a difficult task, especially in complex environments with uneven terrain and various obstacles. We expect a robot to be able to navigate successfully through these environments by using all its limbs for locomotion, in human-like fashion. In general, humanoid robots use the end-effectors (i.e. hands and feet) for making contacts with the environment, but in more difficult situations other parts of the robot body, like the knees or the elbows, can be taken into consideration.

Ideally, a humanoid robot encountering a new environment and given the goal of traversing it, would perform a brief survey to determine the possible contact points, devise a sequence of multi-contact poses based on these contact points, plan a kinematically and dynamically feasible motion, and start moving towards the goal, all within a few seconds and without human supervision. To this end, this paper presents a new dynamic multi-contact motion planner based on the three-dimensional DCM introduced in [1] as an extension of the capture point [2], and the contact force optimization from [3]. We expand on the work in [4] to derive a closed-form solution for the reference CoM trajectory. We take advantage of this property to efficiently compute the exact moments in time for the contact transitions that lead to a feasible trajectory. For executing the motion we integrate the DCM controller into the passivity-based whole-body controller introduced in [5].

All authors are with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany. Email: george.mesesan@dlr.de
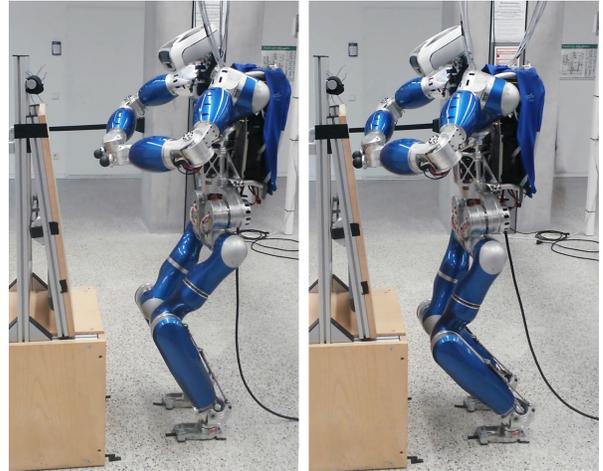
Fig. 1: TORO performing a multi-contact experiment described in Section VI

There are several other works that cover dynamic multi-contact motion. A statistical overview of how humans use multiple contacts to perform various locomotion and manipulation tasks based on motion capture analysis is given in [6]. A dynamic CoM trajectory for a humanoid robot is generated in [7] using a Model Preview Controller, while the contact transition timing is computed using heuristics on the spatial information from the multi-contact planner. A different approach is presented in [8], which uses the virtual-linkage model to provide a physical representation of the CoM resulting force with respect to contact forces and a whole-body controller using task prioritization. A similar method of distributing a given force among predefined contact points providing gravity compensation and adaptation to unknown external forces is given in [9]. The contact forces are transformed into joint torques directly without computing inverse dynamics. An algorithm for testing static equilibrium at particular CoM positions in multi-contact scenarios was presented in [10]. The approach in [11] relies on multi-contact dynamics formulation and derives kinematic trajectories that fulfill dynamic constraints, while in [12] the motion is planned directly in joint space using B-spline time parametrization and nonlinear optimization techniques. In [13], feasible CoM and centroidal momentum trajectories are generated via optimization of contact force base functions and tracked via a linear-quadratic regulator (LQR) controller. A similar approach for trajectory optimization based on centroidal momentum is presented in [14].

II. OVERVIEW

Our approach to solving dynamic multi-contact transitions consists of a *trajectory planner* and a real-time *multi-contact transitions controller* (see Fig.2). Given a sequence of N robot stances $(\sigma_i)_{i=1}^{N}$ generated by an external contact planner, we choose manually a corresponding sequence of step points $(\boldsymbol{\nu}_i)_{i=1}^{N}$, which is used to generate the CoM reference trajectory. The planner automatically computes the timing of the contact transitions and the step durations such that the CoM and end-effector trajectories are kinematically and dynamically feasible with respect to a simplified humanoid robot model. Based on the output of the planner, the controller computes the real-time values of the CoM and end-effector reference trajectories, which are tracked by a passivity-based whole-body controller.

The rest of the paper is organized as follows: in Section III we give a short overview of the Divergent Component of Motion, which serves as a starting point for the closed-form trajectory generation in Section IV. We describe the trajectory planner in Section V, and we present the simulation and experimental results in Section VI, followed by the conclusion in Section VII.

III. DIVERGENT COMPONENT OF MOTION (DCM)

The three-dimensional Divergent Component of Motion (DCM) was defined in [1] as a linear combination of the CoM position $\boldsymbol{x}$ and velocity $\dot{\boldsymbol{x}}$:

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

where b is a time constant defined as $b = \sqrt{\frac{\Delta z_{vrp}}{g}}$. Δz_{vrp} can be interpreted as the average CoM height above the ground surface (see [1] for more details) and g denotes the gravitational constant.

In contrast to other frameworks that typically use the CoM position $\boldsymbol{x}$ and velocity $\dot{\boldsymbol{x}}$ as system states, the DCM framework ([1], [15]) uses the CoM position $\boldsymbol{x}$ and the DCM $\boldsymbol{\xi}$ as system states by applying the following transformation:

$$\begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{\xi} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & b \end{pmatrix} \begin{pmatrix} \boldsymbol{x} \\ \dot{\boldsymbol{x}} \end{pmatrix}. \tag{2}$$

Reordering (1), the CoM dynamics is found to be a stable first order dynamics

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

meaning that the CoM $\boldsymbol{x}$ follows the DCM $\boldsymbol{\xi}$ automatically. Due to its natural stability, it is not necessary to explicitly control the CoM dynamics. In contrast, the DCM dynamics has the following unstable first-order dynamics:

$$\dot{\boldsymbol{\xi}} = \frac{1}{b}\,(\boldsymbol{\xi} - \boldsymbol{\nu}). \tag{4}$$

Here, $\boldsymbol{\nu}$ denotes the Virtual Repellent Point (VRP), which encodes the total force $\boldsymbol{f}$ (including gravity) acting on the CoM via the following relation:

$$\boldsymbol{f} = \frac{m}{b^2}\,(\boldsymbol{x} - \boldsymbol{\nu}), \tag{5}$$

where m denotes the total robot mass. Note that the CoM dynamics is decoupled from the rest of the robot dynamics [16]:

$$\boldsymbol{f} = m\ddot{\boldsymbol{x}}. \tag{6}$$

Due to its unstable nature, the DCM needs to be controlled explicitly. A tracking controller of the form

$$\boldsymbol{\nu} = \underbrace{\boldsymbol{\xi}_d - b\,\dot{\boldsymbol{\xi}}_d}_{\boldsymbol{\nu}_d} + (\boldsymbol{I}_{3\times3} + b\boldsymbol{K}_\xi)(\boldsymbol{\xi} - \boldsymbol{\xi}_d), \tag{7}$$

as proposed in [1], where $\boldsymbol{K}_\xi$ is a diagonal, positive definite matrix, yields the following stable first-order dynamics of the DCM:

$$\underbrace{\dot{\boldsymbol{\xi}} - \dot{\boldsymbol{\xi}}_d}_{\dot{\boldsymbol{e}}_\xi} = -\boldsymbol{K}_\xi \underbrace{(\boldsymbol{\xi} - \boldsymbol{\xi}_d)}_{\boldsymbol{e}_\xi}, \tag{8}$$

i.e., the DCM error $\boldsymbol{e}_\xi$ converges asymptotically.

IV. REFERENCE TRAJECTORIES

One possible application of the DCM framework is the derivation of CoM reference trajectories $\boldsymbol{x}_d, \dot{\boldsymbol{x}}_d, \ddot{\boldsymbol{x}}_d$, which can be tracked by a standard PD feedback controller with feedforward terms. To this end, in this section we derive a reference DCM trajectory $\boldsymbol{\xi}_d$, which can then be transformed into a CoM reference trajectory $\boldsymbol{x}_d$ using the following two equations, which we obtain by applying (3) and (4) to reference quantities (index "d"):

$$\dot{\boldsymbol{x}}_d = -\frac{1}{b}\,(\boldsymbol{x}_d - \boldsymbol{\xi}_d) \tag{9}$$

and

$$\dot{\boldsymbol{\xi}}_d = \frac{1}{b}\,(\boldsymbol{\xi}_d - \boldsymbol{\nu}_d). \tag{10}$$

Given a reference trajectory for $\boldsymbol{\nu}_d$, we can solve the linear system (9)-(10) for $\boldsymbol{x}_d$ and $\boldsymbol{\xi}_d$. We obtain the decoupled reference CoM dynamics by applying (5) and (6) to reference quantities:

$$\boldsymbol{f}_d = m\ddot{\boldsymbol{x}}_d, \tag{11}$$

$$\ddot{\boldsymbol{x}}_d = \frac{1}{b^2}\,(\boldsymbol{x}_d - \boldsymbol{\nu}_d). \tag{12}$$

Finally, combining (5), (7), (9), (10), and (12), we obtain a PD feedback control law with feedforward term for precise tracking:

$$\boldsymbol{f} = \frac{m}{b}\boldsymbol{K}_\xi(\boldsymbol{x}_d - \boldsymbol{x}) + \frac{m}{b}\,(\boldsymbol{I}_{3\times3} + b\boldsymbol{K}_\xi)\,(\dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}}) + m\,\ddot{\boldsymbol{x}}_d. \tag{13}$$

This formulation allows us to integrate the DCM controller into the passivity-based whole-body controller from [5], where the CoM position is controlled by a linear spring-damper system with stiffness matrix $\boldsymbol{K}_c$ and damping $\boldsymbol{D}_c$. For the DCM controller, these matrices are:

$$\begin{aligned} \boldsymbol{K}_c &= \frac{m}{b}\boldsymbol{K}_\xi, \\ \boldsymbol{D}_c &= \frac{m}{b}\,(\boldsymbol{I}_{3\times3} + b\boldsymbol{K}_\xi). \end{aligned} \tag{14}$$
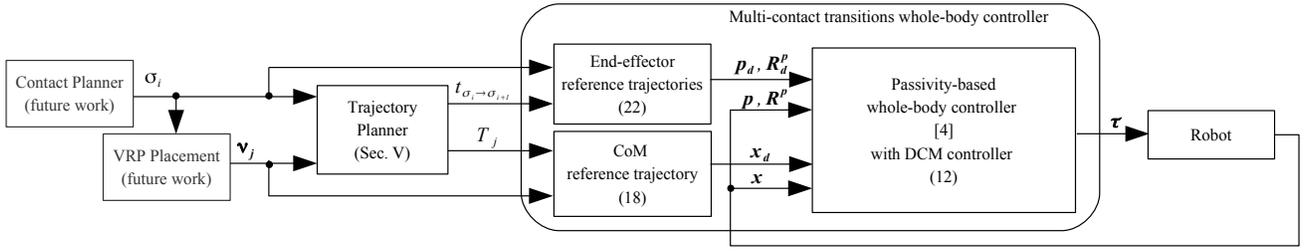
Fig. 2: Overview of the system architecture

A. Computation of single step trajectories

We define a *reference trajectory segment* of duration T as the reference trajectories of the VRP, DCM and CoM ($\boldsymbol{\nu}_d(t)$, $\boldsymbol{\xi}_d(t)$ and $\boldsymbol{x}_d(t)$, respectively) on the time interval $t \in [0, T]$, induced by the initial and final VRP positions ($\boldsymbol{\nu}_{d,0}$ and $\boldsymbol{\nu}_{d,T}$, respectively) and the initial reference system state $(\boldsymbol{x}_{d,0}^T \; \boldsymbol{\xi}_{d,0}^T)^T$.

For the following trajectory computations we introduce a normalized trajectory segment time t_T and the constant b_T:

$$t_T = \frac{t}{T}, \; b_T = \frac{b}{T}.$$

For the VRP reference trajectory, we use the fifth-order polynomial introduced in [4]:

$$\boldsymbol{\nu}_d(t) = \boldsymbol{\nu}_{d,0} + (10 \, t_T^3 - 15 \, t_T^4 + 6 \, t_T^5) \, (\boldsymbol{\nu}_{d,T} - \boldsymbol{\nu}_{d,0}), \; (15)$$

whose main advantages are that it starts and ends with zero velocity and acceleration, and leads to smooth DCM and CoM trajectories, as we show below. Note that this trajectory is a temporal polynomial interpolation, and a spatially linear interpolation, i.e. all points $\boldsymbol{\nu}_d(t)$ are on the line connecting $\boldsymbol{\nu}_{d,0}$ and $\boldsymbol{\nu}_{d,T}$.

We compute the reference DCM trajectory by solving (10) with the boundary condition $\boldsymbol{\xi}_d(0) = \boldsymbol{\xi}_{d,0}$:

$$\begin{aligned}
\boldsymbol{\xi}_d(t) = &\boldsymbol{\nu}_d(t) + b \, \dot{\boldsymbol{\nu}}_d(t) + b^2 \, \ddot{\boldsymbol{\nu}}_d(t) \\
&+ \boldsymbol{\gamma}_1 \frac{t}{b}(t_T + 2 \, b_T - 1) + \boldsymbol{\gamma}_2 + 2\boldsymbol{\gamma}_3 e^{\frac{t}{b}},
\end{aligned} \quad (16)$$

where

$$\begin{aligned}
\boldsymbol{\gamma}_1 &= 360 \, b_T^4 \, (\boldsymbol{\nu}_{d,T} - \boldsymbol{\nu}_{d,0}), \\
\boldsymbol{\gamma}_2 &= \boldsymbol{\gamma}_1 (2 \, b_T - 1) + 60 \, b_T^3 \, (\boldsymbol{\nu}_{d,T} - \boldsymbol{\nu}_{d,0}), \\
\boldsymbol{\gamma}_3 &= \frac{1}{2}(\boldsymbol{\xi}_{d,0} - \boldsymbol{\nu}_{d,0} - \boldsymbol{\gamma}_2)
\end{aligned} \quad (17)$$

are time-independent vectors. The final DCM position of a reference trajectory segment is:

$$\boldsymbol{\xi}_{d,T} = \boldsymbol{\xi}_d(T) = \boldsymbol{\nu}_{d,T} + 2\boldsymbol{\gamma}_1 + \boldsymbol{\gamma}_2 + 2\boldsymbol{\gamma}_3 e^{\frac{T}{b}}. \quad (18)$$

We denote by *fixed-VRP trajectory segment* a trajectory segment with constant VRP position $\boldsymbol{\nu}_{d,T} = \boldsymbol{\nu}_{d,0}$, and by *transition-VRP trajectory segment* a trajectory segment where $\boldsymbol{\nu}_{d,T} \neq \boldsymbol{\nu}_{d,0}$. Note that the DCM moves ($\boldsymbol{\xi}_{d,T} \neq \boldsymbol{\xi}_{d,0}$) even if the VRP is constant; the DCM is constant only if $\boldsymbol{\xi}_{d,0} = \boldsymbol{\nu}_{d,0}$ also holds.

If we choose $\boldsymbol{\xi}_{d,T}$ instead of $\boldsymbol{\xi}_{d,0}$ as boundary condition

for (10), we can compute the initial DCM position as follows:

$$\boldsymbol{\xi}_{d,0} = \boldsymbol{\nu}_{d,0} + \boldsymbol{\gamma}_2 + e^{-\frac{T}{b}}(\boldsymbol{\xi}_{d,T} - \boldsymbol{\nu}_{d,T} - 2\boldsymbol{\gamma}_1 - \boldsymbol{\gamma}_2). \quad (19)$$

Further, we compute the reference CoM trajectory by solving (9) with the boundary condition $\boldsymbol{x}_d(0) = \boldsymbol{x}_{d,0}$:

$$\boldsymbol{x}_d(t) = \boldsymbol{\nu}_d(t) + b^2 \, \ddot{\boldsymbol{\nu}}_d(t) + \boldsymbol{\gamma}_1(2 \, t_T - 1) + \boldsymbol{\gamma}_3 e^{\frac{t}{b}} + \boldsymbol{\gamma}_4 e^{-\frac{t}{b}}, \quad (20)$$

where

$$\boldsymbol{\gamma}_4 = \boldsymbol{\gamma}_1 - \boldsymbol{\gamma}_3 + (\boldsymbol{x}_{d,0} - \boldsymbol{\nu}_{d,0})$$

is a time-independent vector. The final CoM position of a reference trajectory segment is:

$$\boldsymbol{x}_{d,T} = \boldsymbol{x}_d(T) = \boldsymbol{\nu}_{d,T} + \boldsymbol{\gamma}_1 + \boldsymbol{\gamma}_3 e^{\frac{T}{b}} + \boldsymbol{\gamma}_4 e^{-\frac{T}{b}}. \quad (21)$$

B. Computation of trajectories for a sequence of steps

Given an initial reference system state $(\boldsymbol{x}_{d,0}^T \; \boldsymbol{\xi}_{d,0}^T)^T$ and a sequence of N trajectory segments $(T_i, \boldsymbol{\nu}_{d,0,i}, \boldsymbol{\nu}_{d,T,i})_{i=1}^N$ for which the following equalities hold:

$$\boldsymbol{\nu}_{d,0,i} = \boldsymbol{\nu}_{d,T,i-1},$$

we calculate the reference trajectories for VRP, DCM, and CoM on the time interval $t \in [0, T]$, where $T = \sum_{i=1}^N T_i$ is the total duration, by computing the trajectories iteratively for each segment i and taking:

$$\begin{aligned}
\boldsymbol{\xi}_{d,0,i} &= \boldsymbol{\xi}_{d,T,i-1}, \\
\boldsymbol{x}_{d,0,i} &= \boldsymbol{x}_{d,T,i-1}.
\end{aligned}$$

Note that for any trajectory, $\boldsymbol{\nu}_d(t)$ is C^2 continuous, $\boldsymbol{\xi}_d(t)$ is C^3 continuous, while $\boldsymbol{x}_d(t)$ is C^4 continuous.

In most cases, instead of choosing the initial value $\boldsymbol{\xi}_{d,0}$, we prefer to choose a value for $\boldsymbol{\xi}_{d,T,N}$, typically equal to $\boldsymbol{\nu}_{d,T,N}$, meaning that the robot will come to a stop at the final VRP position. We compute $\boldsymbol{\xi}_{d,0}$ by applying (19) recursively, starting with the last step and taking $\boldsymbol{\xi}_{d,T,i} = \boldsymbol{\xi}_{d,0,i+1}$ for $i = 1..N$. For $\boldsymbol{x}_{d,0}$, we choose simply $\boldsymbol{x}_{d,0} = \boldsymbol{x}_0$, where $\boldsymbol{x}_0$ is the initial CoM position.

We use the computed trajectories in two ways:

- For planning purposes, as a preview of the intended trajectory, where we verify the trajectory feasibility with respect to kinematic and dynamic constraints, which we discuss below in Section V;
- As part of the whole-body controller from Fig. 2, where we compute in real-time the values for $\boldsymbol{x_d}$, $\dot{\boldsymbol{x}}_d$ and $\ddot{\boldsymbol{x}}_d$.

V. Trajectory planning

Our goal is to plan kinematically and dynamically feasible whole-body trajectories that can be tracked by the real-time controller. To reduce planning time, we use a simplified humanoid robot model (Fig. 3). Kinematically, it consists of a cuboid body with limbs defined as 3 DOF manipulators of RRP type (two rotational and one prismatic joint), with the first rotation axis along the y-axis, the second along the rotated x-axis, and the prismatic joint in the negative direction of the resulting z-axis. The end-effector orientation can be chosen arbitrarily such that it matches the orientation of the contact surface. Dynamically, we consider the limbs to be massless and we ignore the rotational inertia of the body, which is equivalent to a point-mass model with a fixed CoM position relative to the cuboid body. The methods from the previous sections are used to efficiently generate CoM trajectories and corresponding force profiles and to verify their feasibility. We rely on the robustness of the controller to compensate for the simplifying assumptions made regarding the kinematic and dynamic models.
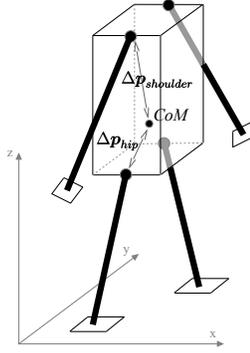

Fig. 3: Simplified humanoid robot model used for planning

A. Problem formulation

We define a *contact* as the tuple $c = (\boldsymbol{p}_c, \boldsymbol{R}_c)$, where $\boldsymbol{p}_c \in \mathbb{R}^3$ is the point of contact in world coordinates, and $\boldsymbol{R}_c \in SO(3)$ is the orientation of the contact frame, chosen such that the z-axis is aligned with the contact surface normal. A *stance* [17] is a set of K contacts $\sigma = \{c_k \mid k = 1..K\}$. For two stances σ_i and σ_{i+1} we denote by $t_{\sigma_i \rightarrow \sigma_{i+1}}$ the moment in time when the *transition* from σ_i to σ_{i+1} takes place. For a contact c there are three transition cases:

- *attach*: $c \notin \sigma_i$ and $c \in \sigma_{i+1}$;
- *detach*: $c \in \sigma_i$ and $c \notin \sigma_{i+1}$;
- *hold*: $c \in \sigma_i$ and $c \in \sigma_{i+1}$.

Given a sequence of N stances $(\sigma_i)_{i=1}^N$, the initial system state $(\boldsymbol{x}_0^T \ \boldsymbol{\xi}_0^T)^T$, and the final reference system state $(\boldsymbol{x}_{d,T}^T \ \boldsymbol{\xi}_{d,T}^T)^T$, we aim to find a reference CoM trajectory $\boldsymbol{x}_d(t), \dot{\boldsymbol{x}}_d(t), \ddot{\boldsymbol{x}}_d(t)$ of total duration T, as well as the contact transition times $t_{\sigma_i \rightarrow \sigma_{i+1}}$, for $i = 1..N$, such that the motion fulfills kinematic and dynamic constraints.

B. Kinematic feasibility

We verify kinematic feasibility by computing reference trajectories for the limb end-effectors, solving the inverse kinematics for each limb and verifying that the joint limits are not violated. If the contact is held at $\boldsymbol{p}_c$, the end-effector reference position is $\boldsymbol{p}_d(t) = \boldsymbol{p}_c$. In contrast, if a limb end-effector is detached from a contact point $\boldsymbol{p}_{c_i}$ at time t_i, and later attached to a contact point $\boldsymbol{p}_{c_j}$ at time t_j, then we interpolate the limb end-effector reference position $\boldsymbol{p}_d(t)$ using the fifth-order polynomial:

$$\boldsymbol{p}_d(t) = \boldsymbol{p}_{c_i} + (10 \, t_{T_c}^3 - 15 \, t_{T_c}^4 + 6 \, t_{T_c}^5) \, (\boldsymbol{p}_{c_j} - \boldsymbol{p}_{c_i}), \quad (22)$$

where $t_{T_c} = t/T_c$, and $T_c = t_j - t_i$ is the duration of the end-effector motion. Note that this is the same interpolation as (15), so the generated trajectory has the same properties. We impose a maximum velocity constraint $\boldsymbol{v}_{max}$ on $\dot{\boldsymbol{p}}_d$, which allows us to compute a minimum bound on the duration T_c:

$$T_{c,min} = \frac{15}{8\boldsymbol{v}_{max}} |\boldsymbol{p}_{c_j} - \boldsymbol{p}_{c_i}|,$$

which we obtained by finding that $\boldsymbol{v}_{max} = \dot{\boldsymbol{p}}_{d,max} = \dot{\boldsymbol{p}}_d(T/2)$. For a reference CoM position $\boldsymbol{x}_d$ and the limb configuration $\boldsymbol{q} = (\alpha, \beta, d)$, we compute the end-effector position $\boldsymbol{p}$ as follows:

$$\boldsymbol{p} = \boldsymbol{x}_d + \Delta\boldsymbol{p}_{joint} + \begin{pmatrix} -d \, sin(\alpha)cos(\beta) \\ d \, sin(\beta) \\ -d \, cos(\alpha)cos(\beta) \end{pmatrix}, \quad (23)$$

where $\Delta\boldsymbol{p}_{joint}$ is the position of the RRP joint relative to the CoM ($\Delta\boldsymbol{p}_{hip}$ for legs, and $\Delta\boldsymbol{p}_{shoulder}$ for arms). The body is kept upright for the whole duration of the motion; changing the body orientation to achieve feasibility or to improve it by moving away from joint limits is part of our ongoing research.

A RRP limb configuration $\boldsymbol{q}$ is *feasible* if it is within the joint limits: $\boldsymbol{q}_{min} \leqslant \boldsymbol{q} \leqslant \boldsymbol{q}_{max}$. The reference CoM trajectory $\boldsymbol{x}_d(t)$ is *kinematically feasible* for the reference end-effector trajectories $\boldsymbol{p}_d(t)$ if we can find feasible configurations $\boldsymbol{q}(t)$ for all limbs such that $\boldsymbol{p}(t) = \boldsymbol{p}_d(t)$, i.e. solving the inverse kinematics. Note that for $\beta = \pm 90°$ in (23), we cannot determine the angle α; in this case, we ignore its value for the feasibility check. In practice, we verify kinematic feasibility by discretizing the trajectory with a time step Δt.

C. Dynamic feasibility

A contact wrench $\boldsymbol{w}_c = (f_x \ f_y \ f_z \ \tau_x \ \tau_y \ \tau_z)^T$, expressed in the contact frame, acting at the contact point $\boldsymbol{p}_c$ is *feasible* if it fulfills the following constraints:

- contact unilaterality: $f_z > 0$;
- force is within the friction cone. We use a polyhedral approximation of the friction cone, so that we have linear constraints for the tangential forces: $|f_x| < \mu f_z$ and $|f_y| < \mu f_z$, where μ is the friction coefficient;
- maximum normal force: $f_z < f_{z,max}$;
- minimum normal force: $f_{z,min} < f_z$, ensuring that the contact remains stable;
- maximum torque on z-axis: $|\tau_z| < \tau_{z,max}$;

- Center of Pressure (CoP) is within the contact surface polygon S: $\boldsymbol{p}_{CoP} = (-\frac{\tau_y}{f_z}, \frac{\tau_x}{f_z})^T \in S$.

A wrench $\boldsymbol{w}_c$ applied at the contact point can be transformed into an equivalent wrench $\boldsymbol{w}_{d,c}$ acting at the reference CoM position $\boldsymbol{x}_d$ using the following transformation:

$$\boldsymbol{w}_{d,c} = \underbrace{\begin{pmatrix} \boldsymbol{R}_c & \boldsymbol{0}_{3\times3} \\ [(\boldsymbol{p}_c - \boldsymbol{x}_d)\times]\boldsymbol{R}_c & \boldsymbol{R}_c \end{pmatrix}}_{\boldsymbol{G}_k} \boldsymbol{w}_c.$$

For a stance σ with K contacts, the total wrench acting on the CoM is

$$\boldsymbol{w}_{d,\sigma} = \sum_{k=1}^{K} \boldsymbol{G}_k \boldsymbol{w}_{c_k} = \boldsymbol{G}\,\boldsymbol{w}_\sigma,$$

where $\boldsymbol{G} = (\boldsymbol{G}_1 \dots \boldsymbol{G}_K)$ and $\boldsymbol{w}_\sigma = (\boldsymbol{w}_{c_1}^T \dots \boldsymbol{w}_{c_K}^T)^T$. Given a reference wrench $\boldsymbol{w}_d$, we verify its feasibility for the stance σ by solving the constrained quadratic optimization problem (QP):

$$\min_{\boldsymbol{w}_\sigma} \frac{1}{2}(\boldsymbol{G}\,\boldsymbol{w}_\sigma - \boldsymbol{w}_d)^T \boldsymbol{Q}(\boldsymbol{G}\,\boldsymbol{w}_\sigma - \boldsymbol{w}_d),$$

such that $\boldsymbol{w}_\sigma$ is composed of feasible contact wrenches. Here, $\boldsymbol{Q}$ is a symmetric and positive definite weighting matrix. The wrench $\boldsymbol{w}_d$ is *feasible* for the stance σ if the magnitude of the divergent wrench vector

$$\boldsymbol{w}_{div,\sigma} = \boldsymbol{G}\,\boldsymbol{w}_\sigma - \boldsymbol{w}_d$$

is less than the tolerance value $w_{div,max}$.

We define a reference CoM trajectory $\boldsymbol{x}_d(t), \dot{\boldsymbol{x}}_d(t), \ddot{\boldsymbol{x}}_d(t)$ to be *dynamically feasible* for the stance σ if all wrenches $\boldsymbol{w}_d(t) = (\boldsymbol{f}_{d,g}^T(t)\ \boldsymbol{0}_{1\times3})^T$ are feasible, where $\boldsymbol{f}_{d,g}(t) = m\ddot{\boldsymbol{x}}_d(t) - m\boldsymbol{g}$, with $\boldsymbol{g} = (0\ 0\ -g)^T$ being the gravitational acceleration. Note that the reference CoM position $\boldsymbol{x}_d(t)$ is also part of the computation, as it is included in all $\boldsymbol{G}_k$. In practice, we verify the dynamic feasibility by discretizing the trajectory with the same time step Δt as for kinematic feasibility, and solving a series a QPs accordingly.

D. Planning algorithm

Given a sequence of N stances $(\sigma_i)_{i=1}^N$, we start by choosing manually reference VRP positions $\boldsymbol{\nu}_{d,i}$ for each stance σ_i. Intuitively, we prefer VRP positions which can act as CoM rest positions, i.e. a gravity compensation wrench $\boldsymbol{w}_d = (-m\boldsymbol{g}^T\ \boldsymbol{0}_{1\times3})^T$ is feasible. However, depending on the stance and the contact surface properties (e.g. slippery inclines), this might not always be possible. An automatic method of finding VRP positions that lead to feasible trajectories is part of our ongoing research efforts.

Based on the previously mentioned reference VRP positions, we create a reference trajectory plan by alternating between fixed-VRP and transition-VRP trajectory segments: $\boldsymbol{\nu}_{d,1}, \boldsymbol{\nu}_{d,1}, \boldsymbol{\nu}_{d,2}, \boldsymbol{\nu}_{d,2}, \dots, \boldsymbol{\nu}_{d,N}, \boldsymbol{\nu}_{d,N}$, for a total number of steps $M = 2N - 1$. Now, the main advantage of VRP-based motion becomes apparent, as the search for feasible CoM trajectories $\boldsymbol{x}_d(t), \dot{\boldsymbol{x}}_d(t), \ddot{\boldsymbol{x}}_d(t)$ is replaced by the search for the step durations $(T_j)_{j=1}^M$ that together with the VRP sequence above induces feasible trajectories; this search

problem is comparatively easier to solve, as we show below. An alternative VRP-motion plan is to use only transition trajectory segments (N steps), however, in general, this type of plan leads to a longer total duration. The main difference lies in the magnitude of the reference CoM accelerations (12), which are higher in the case of fixed-VRP segments.

For two stances σ_i and σ_{i+1}, the contact transition time can be found in the transition-VRP trajectory segment $\boldsymbol{\nu}_i \to \boldsymbol{\nu}_{i+1}$ (see Fig. 4) using a binary search on the interval $[t_{\boldsymbol{\nu}_{d,T,i}}, t_{\boldsymbol{\nu}_{d,0,i+1}}]$. We are searching for the time $t_{\sigma_i \to \sigma_{i+1}}$ for which the wrench $\boldsymbol{w}_d$ is feasible for both stances σ_i and σ_{i+1} (see Algorithm 1 for the details). If a solution is found, we check the dynamic feasibility for the stance σ_i on the interval $[t_{\boldsymbol{\nu}_{d,T,i}}, t_{\sigma_i \to \sigma_{i+1}}]$ and for σ_{i+1} on the interval $[t_{\sigma_i \to \sigma_{i+1}}, t_{\boldsymbol{\nu}_{d,0,i+1}}]$. If no solution is found, then the trajectory is unfeasible.

Algorithm 1 Finding the contact transition time

1: $t_{min} \leftarrow t_{\boldsymbol{\nu}_{d,T,i}}$
2: $t_{max} \leftarrow t_{\boldsymbol{\nu}_{d,0,i+1}}$
3: **repeat**
4:     $t_c \leftarrow (t_{min} + t_{max})/2$
5:     compute $\boldsymbol{x}_d(t_c), \dot{\boldsymbol{x}}_d(t_c), \ddot{\boldsymbol{x}}_d(t_c)$
6:     $\boldsymbol{f}_{d,g} \leftarrow m\ddot{\boldsymbol{x}}_d(t_c) - m\boldsymbol{g}$
7:     $\boldsymbol{w}_d \leftarrow (\boldsymbol{f}_{d,g}^T\ \boldsymbol{0}_{1\times3})^T$
8:     **if** *feasible*$(\boldsymbol{w}_d, \sigma_i) \wedge$ *feasible*$(\boldsymbol{w}_d, \sigma_{i+1})$ **then**
9:         **return** t_c
10:     **else if** $\neg$*feasible*$(\boldsymbol{w}_d, \sigma_i) \wedge$ *feasible*$(\boldsymbol{w}_d, \sigma_{i+1})$ **then**
11:         $t_{max} \leftarrow t_c$
12:     **else if** *feasible*$(\boldsymbol{w}_d, \sigma_i) \wedge \neg$*feasible*$(\boldsymbol{w}_d, \sigma_{i+1})$ **then**
13:         $t_{min} \leftarrow t_c$
14:     **else if** $\neg$*feasible*$(\boldsymbol{w}_d, \sigma_i) \wedge \neg$*feasible*$(\boldsymbol{w}_d, \sigma_{i+1})$ **then**
15:         **return** failure
16:     **end if**
17: **until** $t_{max} - t_{min} < \Delta t$

An initial sequence of feasible step durations $(T_j)_{j=1}^M$ for the reference trajectory segments (the bottom part of the plan in Fig. 4), is found by using an iterative randomized search with multiple candidates on the space $\mathbb{R}_+^M$, with initial guesses for T_j taken from the interval $[T_{min}, T_{max}]$. T_{min} and T_{max} are design parameters that, in our experience, are typical lower and upper bounds for step duration in DCM-based motion. We guide the search by selecting and making small adjustments to the most promising candidates, using the count of unfeasible points as a measure for the distance to the feasible subspace. The search terminates when the first feasible solution $T_{initial}$ is found or when a maximum number of iterations I_{max} is reached, in which case we conclude that the trajectory is unfeasible. We improve the solution with Algorithm 2 to obtain a faster executing reference trajectory plan $T_{feasible}$ which is executed by the real-time controller. The algorithm iterates over all trajectory segments and verifies if reducing the duration of a trajectory segment preserves the feasibility of the whole trajectory.
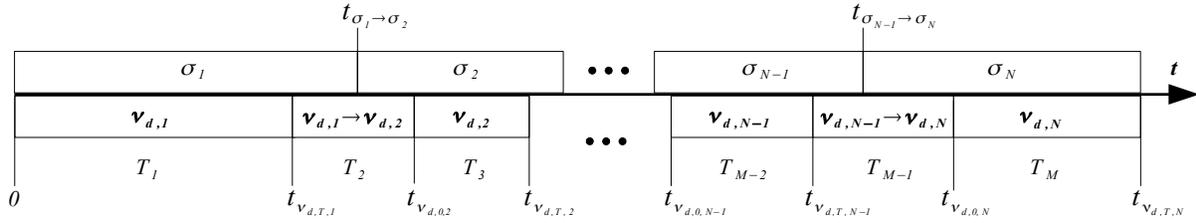
$t_{\sigma_1 \to \sigma_2}$ $t_{\sigma_{N-1} \to \sigma_N}$

| σ_1 | σ_2 | $\cdots$ | σ_{N-1} | σ_N | t |

| $\boldsymbol{v}_{d,1}$ | $\boldsymbol{v}_{d,1} \to \boldsymbol{v}_{d,2}$ | $\boldsymbol{v}_{d,2}$ | $\cdots$ | $\boldsymbol{v}_{d,N-1}$ | $\boldsymbol{v}_{d,N-1} \to \boldsymbol{v}_{d,N}$ | $\boldsymbol{v}_{d,N}$ |

| T_1 | T_2 | T_3 | | T_{M-2} | T_{M-1} | T_M |

0 $t_{v_{d,T,1}}$ $t_{v_{d,0,2}}$ $t_{v_{d,T,2}}$ $t_{v_{d,0,N-1}}$ $t_{v_{d,T,N-1}}$ $t_{v_{d,0,N}}$ $t_{v_{d,T,N}}$

Fig. 4: Multi-contact transitions plan

Algorithm 2 Improving the reference trajectory plan

1: **for** $j = 1$ to M **do**
2:      $T_{j,min} \leftarrow T_{min}$
3:      $T_{j,max} \leftarrow T_{j,initial}$
4:      $T_{j,feasible} \leftarrow T_{j,initial}$
5:      **repeat**
6:          $T_j \leftarrow (T_{j,min} + T_{j,max})/2$
7:          $T = \sum_{j=1}^{M} T_j$
8:          compute $\boldsymbol{x}_d(t), \dot{\boldsymbol{x}}_d(t), \ddot{\boldsymbol{x}}_d(t)$ for $t \in [0, T]$
9:          **for** $i = 1$ to $N - 1$ **do**
10:              find $t_{\sigma_i \to \sigma_{i+1}}$ with Algorithm 1
11:          **end for**
12:          compute $\boldsymbol{p}_d(t)$ for $t \in [0, T]$
13:          **if** $feasible(\boldsymbol{x}_d(t), \dot{\boldsymbol{x}}_d(t), \ddot{\boldsymbol{x}}_d(t))$ **then**
14:              $T_{j,feasible} \leftarrow T_j$
15:              $T_{j,max} \leftarrow T_j$
16:          **else**
17:              $T_{j,min} \leftarrow T_j$
18:          **end if**
19:      **until** $T_{j,max} - T_{j,min} < \Delta t$
20: **end for**

TABLE I: Parameter values for the trajectory planner

Parameter	Value	
Δz_{vrp}	0.9	[m]
Δt	0.01	[s]
T_{min}	0.2	[s]
T_{max}	10	[s]
$w_{div,max}$	1	
I_{max}	100	
$\boldsymbol{Q}$	diag($10^2\ 10^2\ 10^2\ 10^3\ 10^3\ 10^3$)	

Parameter	FootR, FootL	HandR, HandL	
Δ_{joint}	$(0 \mp 0.113\ 0)$	$(0 \mp 0.25\ 0.45)$	[m]
$\boldsymbol{q}_{min}$	$(-45°\ -20°\ 0.7)$	$(-180°\ -90°\ 0.25)$	
$\boldsymbol{q}_{max}$	$(60°\ 45°\ 1.1)$	$(180°\ 30°\ 0.8)$	
$\boldsymbol{v}_{max}$	$(0.5\ 0.5\ 0.5)$	$(0.75\ 0.75\ 0.75)$	[m/s]
μ	0.4	0.4	
$f_{z,max}$	900	50	[N]
$f_{z,min}$	50	20	[N]
$\tau_{z,max}$	9	5	[Nm]
$\boldsymbol{p}_{CoP}^{min}$	$(-0.05\ -0.035)$	$(-0.025\ -0.025)$	[m]
$\boldsymbol{p}_{CoP}^{max}$	$(0.11\ 0.035)$	$(0.025\ 0.025)$	[m]

VI. EXPERIMENTS

To demonstrate the performance of the methods presented in this paper, we have tested two scenarios in simulation using OpenHRP [18] and performed one experiment with TORO [19], a 27-DoFs humanoid robot developed by DLR (German Aerospace Center), with a height of 1.74 m and a weight of 76.4 kg. For the trajectory planner we used the parameters in Table I, and for the DCM controller we used $\boldsymbol{K}_\xi = \text{diag}(0.3\ 0.3\ 0.3)$; the passivity-based whole-body controller parameters are given in [5].

The first scenario (Fig. 5) begins with TORO standing in front of a wall at a distance of 62 cm. During the experiment the robot leans towards the wall and makes contact with the hands, gradually stopping the forward motion. It holds the final position for 2 seconds (Fig. 1), using the four contact points for balancing, then returns to the initial posture. The contact transition times are computed automatically using Algorithm 1. The CoM reference trajectory is composed of three trajectory segments, whose durations were chosen manually such that the CoM clearly leaves the support polygon. Note that due to the nature of the demonstration (i.e. returing to the initial pose), refining the motion using Algorithm 2 leads to a small forward shift of the CoM from the initial position that fulfills the kinematic requirements of reaching the wall, but does not leave the support polygon.

In the second scenario (Fig. 6), the robots uses the left hand for support while stepping over a large spherical obstacle forwards and backwards. The motion plan consists of 7 stances and 14 reference trajectory segments. The radius of the obstacle is 15 cm, requiring the robot to take a step which is 55 cm long (accounting also for the length of the foot) and 20 cm high. The planner found an initial feasible solution with a total duration of 65.4 s, applying Algorithm 2 reduced the duration to 7.92 s. The planner execution time for finding the initial solution was 0.345 s, and 2.764 s for improving the trajectory.

## VII. CONCLUSION AND FUTURE WORK

We have presented in this paper a new approach to planning and control of dynamic multi-contact transitions for humanoid robots. The main focus was on creating a very fast planner which takes a sequence of robot stances together with corresponding Virtual Repellent Points as input, and produces reference CoM and end-effector trajectories that fulfill kinematic and dynamic constraints. We have achieved this goal by taking advantage of closed-form solutions for
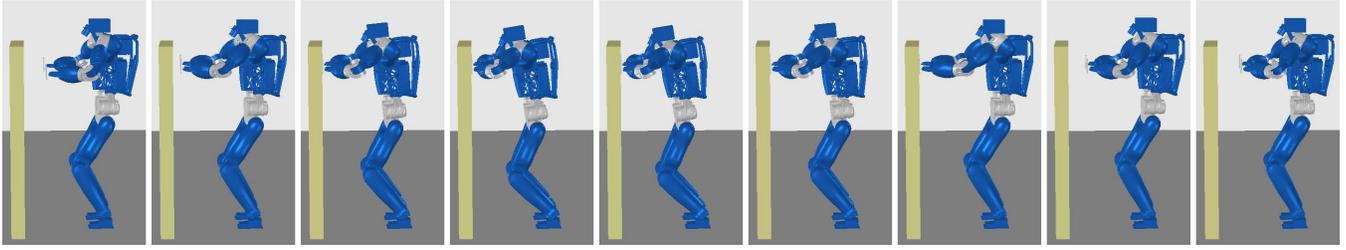
(a) Simulation images (1 second interval)
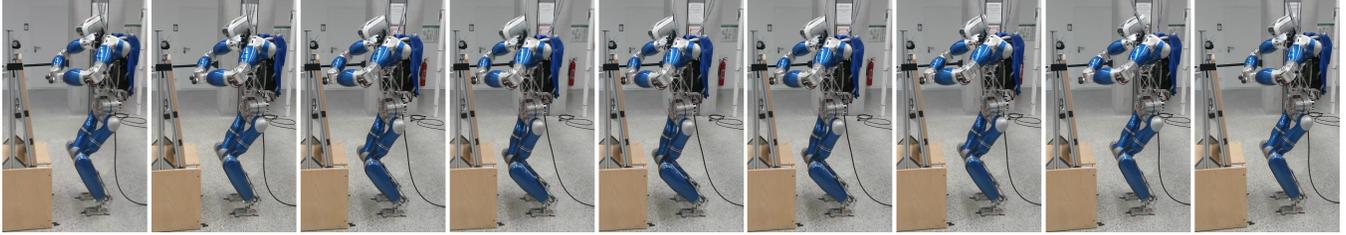
(b) Experiment images (1 second interval)
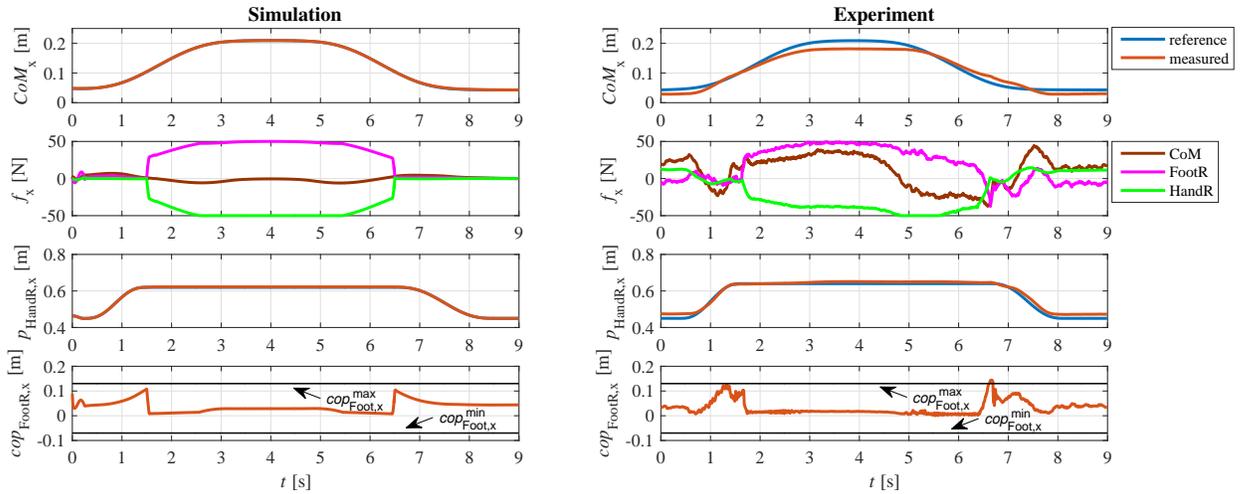
(c) Relevant signals

Fig. 5: Leaning forward and using the wall to support the weight

reference CoM trajectories based on the concept of the Divergent Component of Motion (DCM), as well as by using a simplified humanoid robot model. We have also shown how to combine DCM control with a passivity-based whole-body controller, which allows the tracking of the reference trajectories. We have shown the robustness of the planning solution and the tracking performance of the controller in simulation and in experiments with the humanoid robot TORO.

We will focus our future research efforts on automatic VRP placement and expansion of the robot model used for planning by introducing rotational inertia in the body and replacing the massless limbs with point masses at the end-effector positions. These model improvements will bring the additional challenge of planning the body posture and controlling the angular momentum. Additionally, we will work towards the goal stated in the introduction of achieving fast, autonomous, vision-based planning and robust control of humanoid locomotion in challenging environments.

REFERENCES

[1] 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.

[2] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2006, pp. 200–207.

[3] C. Ott, M. A. Roa, and G. Hirzinger, "Posture and balance control for biped robots based on contact force optimization," in *IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2011, pp. 26–33.

[4] J. Englsberger, "Combining reduced dynamics models and whole-body control for agile humanoid locomotion," Ph.D. dissertation, Technische Universität München, 2016.
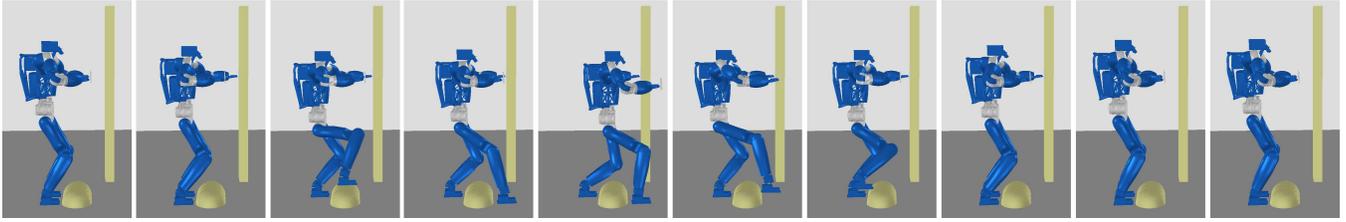
(a) Simulation images (1 second interval)
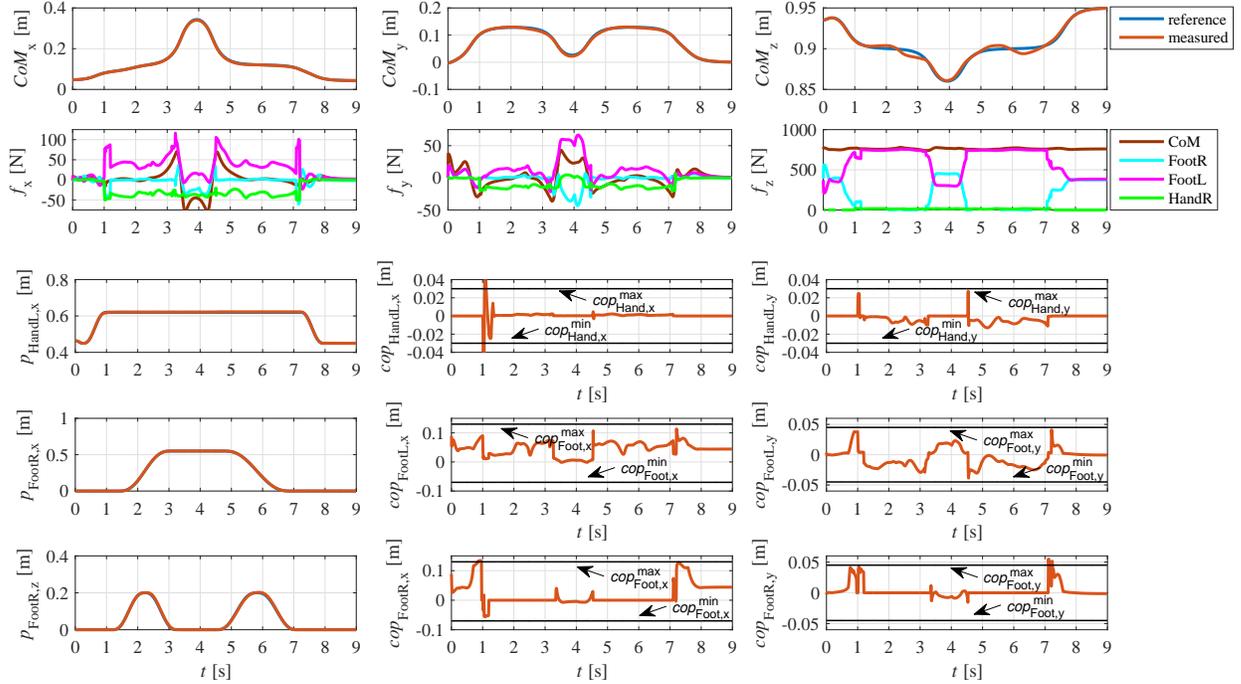
(b) Relevant signals

Fig. 6: Stepping over a large obstacle using one hand for support

[5] 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.

[6] C. Mandery, J. Borras, M. Jochner, and T. Asfour, "Analyzing whole-body pose transitions in multi-contact motions," in *IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2015, pp. 1020–1027.

[7] H. Audren, J. Vaillant, A. Kheddar, A. Escande, K. Kaneko, and E. Yoshida, "Model preview control in multi-contact motion-application to a humanoid robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2014, pp. 4030–4035.

[8] L. Sentis, J. Park, and O. Khatib, "Compliant control of multicontact and center-of-mass behaviors in humanoid robots," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 483–501, 2010.

[9] S.-H. Hyon, J. G. Hale, and G. Cheng, "Full-body compliant human–humanoid interaction: balancing in the presence of unknown external forces," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 884–898, 2007.

[10] T. Bretl and S. Lall, "Testing static equilibrium for legged robots," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 794–807, 2008.

[11] L. Sentis and M. Slovich, "Motion planning of extreme locomotion maneuvers using multi-contact dynamics and numerical integration," in *IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2011, pp. 760–767.

[12] S. Lengagne, J. Vaillant, E. Yoshida, and A. Kheddar, "Generation of whole-body optimal dynamic multi-contact motions," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1104–1119, 2013.

[13] A. Herzog, N. Rotella, S. Schaal, and L. Righetti, "Trajectory generation for multi-contact momentum control," in *IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2015, pp. 874–880.

[14] 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.

[15] T. Takenaka, T. Matsumoto, and T. Yoshiike, "Real time motion generation and control for biped robot, 1st report: Walking gait pattern generation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009.

[16] S. Dubowsky and E. Papadopoulos, "The kinematics, dynamics, and control of free-flying and free-floating space robotic systems," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 5, pp. 531–543, 1993.

[17] K. Bouyarmane and A. Kheddar, "Humanoid robot locomotion and manipulation step planning," *Advanced Robotics*, vol. 26, no. 10, pp. 1099–1126, 2012.

[18] F. Kanehiro, H. Hirukawa, and S. Kajita, "Openhrp: Open architecture humanoid robotics platform," *The International Journal of Robotics Research*, vol. 23, no. 2, pp. 155–165, 2004.

[19] J. Englsberger, A. Werner, C. Ott, B. Henze, M. A. Roa, G. Garofalo, R. Burger, A. Beyer, O. Eiberger, K. Schmid *et al.*, "Overview of the torque-controlled humanoid robot toro," in *IEEE-RAS International Conference on Humanoid Robots*, 2014.