

This is the author's copy of the publication as archived with the DLR's electronic library at <http://elib.dlr.de>. Please consult the original publication for citation.

Online Centroidal Angular Momentum Reference Generation and Motion Optimization for Humanoid Push Recovery

Schuller, Robert; Mesesan, George; Engelsberger, Johannes; Lee, Jinoh; Ott, Christian

Copyright Notice

©2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Citation Notice

```
@ARTICLE{9435940,
  author={Schuller, Robert and Mesesan, George and Engelsberger, Johannes and Lee, Jinoh and Ott, Christian},
  journal={IEEE Robotics and Automation Letters},
  title={Online Centroidal Angular Momentum Reference Generation and Motion Optimization for Humanoid Push Recovery},
  year={2021},
  volume={6},
  number={3},
  pages={5689-5696},
  doi={10.1109/LRA.2021.3082023}
}
```

Online Centroidal Angular Momentum Reference Generation and Motion Optimization for Humanoid Push Recovery

Robert Schuller, George Mesesan, Johannes Engelsberger, Jinhoo Lee, and Christian Ott

Abstract—This paper presents a new push recovery algorithm for humanoid robots in balancing scenarios by exploiting the system’s rotational dynamics. The proposed framework actively generates centroidal angular momentum (CAM) references based on the force magnitude and direction of the push to counteract the disturbance and maintain its balance. Since a humanoid robot can only store a limited amount of angular momentum, the CAM reference is generated in three consecutive phases: 1) CAM generation phase to counteract the push; 2) CAM reduction phase to bring the robot to a halt; and 3) posture recovery phase to converge to the reference pose again. A subsequent whole-body motion optimizer, formulated as a constrained quadratic optimization problem, generates kinematically feasible whole-body trajectories based on the CAM reference. The proposed framework is validated through experiments with the humanoid robot TORO.

I. INTRODUCTION

Maintaining balance under strong external forces even in confined spaces is one of the key challenges for humanoid robots to be deployed in real-world applications and interact with humans. The robot needs to actively react to unpredictable external pushes, i.e., push recovery, to avoid falling, which is always a severe danger for the robot as well as adjacent humans.

To create sophisticated motion in humanoid robots, it is beneficial to analyze human behavior first, since humanoid robots mimic real human beings concerning size and kinematic structure, and thus both share similar motion patterns. Biomechanical studies of human balance [1] discovered different high-level reaction strategies in the presence of external perturbations (ankle, hip, and stepping strategies). The ankle strategy generates torques in the ankle joints while keeping the rest of the body in a fixed posture. Using the hip strategy, humans purposely change the body’s angular momentum to generate horizontal restoring forces acting on the center of mass (CoM). Finally, one can also take a step to withstand a push, called stepping strategy. These strategies are selected by several factors such as the intensity of the perturbation or properties of the support surface [2]. For balancing on confined or compliant ground, a combination of ankle and hip strategy is mainly preferred [3], which is also the focus of this work as previewed in Fig. 1.

Inspired by the importance of angular momentum in human motion, especially expressed around the CoM [4], [5],

This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No. 819358).

All authors are with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany robert.schuller@dlr.de.

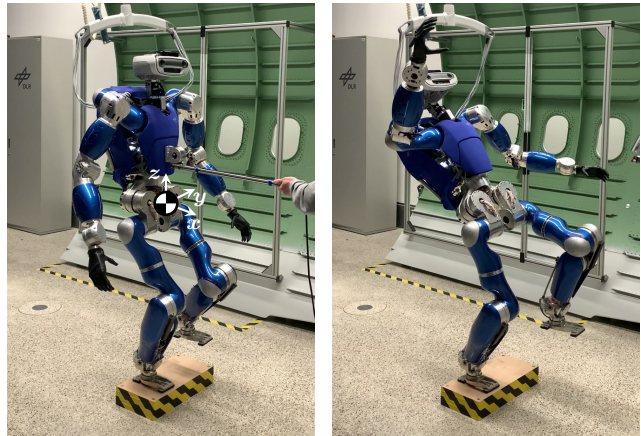


Fig. 1. Humanoid robot TORO generating centroidal angular momentum to maintain its balance after being pushed.

i.e., centroidal angular momentum (CAM), explicit control of the rotational dynamics has become a core component in humanoid whole-body control [6], [7]. Some approaches use simplified models to account for angular momentum in humanoids, e.g., the linear inverted pendulum plus flywheel model [8], [9]. Other methods exploit the robot’s full system dynamics since the simplified models cannot describe the entire complexity of individual body segments’ contributions to the overall CAM of a humanoid [10]–[12].

Especially for balancing controllers that support a hip strategy, where upper body motion generates CAM to withstand the push, a full system model is beneficial. Hyon et al. [13] use a passivity-based approach for their postural balancer combining an ankle and hip strategy to withstand external pushes. Hofmann et al. [14] present a momentum-exploiting CoM controller, which sacrifices angular momentum regulation for a higher prioritized linear momentum task under large external perturbations. The momentum-based balance controller proposed by Lee et al. [15] generates admissible momenta while considering contact constraints. The approach is extended and experimentally validated in [16].

Methods for generating CAM references based on desired motions, e.g., jumps with twist or kicks, are introduced in [17]–[19]. However, the presented approaches [10]–[19] have in common that the CAM for force-disturbed balancing scenarios is not deliberately planned but only generated when the fulfillment of higher priority tasks requires a violation of the angular momentum regulation task. This can be the case when contact constraints are reached, e.g., the center of pressure (CoP) is approaching the edge of the support

area. In contrast to flywheels, humanoids do not have the capability of storing CAM for a longer period of time. If the induced angular momentum is not reduced quickly enough, kinematic limits will be reached [10]. However, if the angular momentum is reduced too quickly, the CoP will move again towards the edge of the support area. In the previous approaches, the behavior of the angular momentum can only be influenced indirectly by reprioritizing different subtasks, which makes it difficult to properly schedule the addition and removal of CAM to the system without reaching hardware or contact constraints.

The main contributions of this work are as follows: (i) we present a push recovery algorithm for force-disturbed balancing scenarios exploiting an ankle and hip strategy, while a corresponding CAM reference is generated online based on the force magnitude and direction of the push. The high-dimensional tuning process of different task priorities to schedule the CAM behavior is simplified by introducing physically intuitive design parameters. The hip strategy can be independently activated even before contact constraints are activated or task conflicts occur; and (ii) we propose a framework that takes the CAM reference as an input and generates feasible whole-body motions while considering kinematic limits of the system and exploiting the full-body dynamics. A constraint quadratic optimization problem (QP) formulation gives the flexibility of weighting the contribution of different body segments to the overall CAM and enables the online adaption to different contact configurations.

The generated whole-body trajectories are tracked by a subsequent passivity-based whole-body controller. Fig. 2 overviews the proposed system architecture.

II. BACKGROUND

This section gives an overview of the system's dynamic model and a summary of the previous work on the passivity-based whole-body controller [20]–[22].

A. Dynamic Model

As a system model for a humanoid robot, we assume a floating base dynamics with n torque-controlled joints. Instead of the base coordinates, we utilize the position of the CoM $\mathbf{x}_c \in \mathbb{R}^3$ together with the orientation of the hip $\mathbf{R}_b \in SO(3)$, and with the corresponding translational and rotational velocities $\dot{\mathbf{x}}_c \in \mathbb{R}^3$ and $\boldsymbol{\omega}_b \in \mathbb{R}^3$, which are stacked together into the velocity vector $\boldsymbol{\nu}_c = (\dot{\mathbf{x}}_c^T \boldsymbol{\omega}_b^T)^T$. The total number of degrees of freedom (DoF) of the system is denoted by $\bar{n} = n + 6$. The dynamics of the system can be formulated as follows:

$$\mathbf{M} \begin{pmatrix} \dot{\boldsymbol{\nu}}_c \\ \Delta \dot{\mathbf{q}} \end{pmatrix} + \mathbf{C} \begin{pmatrix} \boldsymbol{\nu}_c \\ \dot{\mathbf{q}} \end{pmatrix} + \begin{pmatrix} -\mathbf{w}_g \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \boldsymbol{\tau} \end{pmatrix} + \boldsymbol{\tau}_{ext}, \quad (1)$$

where $\mathbf{M} \in \mathbb{R}^{\bar{n} \times \bar{n}}$ and $\mathbf{C} \in \mathbb{R}^{\bar{n} \times \bar{n}}$ are the positive definite inertia and the Coriolis matrix, respectively. The gravitational wrench is represented by $\mathbf{w}_g = (m\mathbf{g}_0^T \mathbf{0}^T)^T$, where m denotes the overall mass of the robot and $\mathbf{g}_0 \in \mathbb{R}^3$ is the vector of gravitational acceleration. The joint positions are represented by $\mathbf{q} \in \mathbb{R}^n$, and the corresponding joint

torques are $\boldsymbol{\tau} \in \mathbb{R}^n$. The variable $\boldsymbol{\tau}_{ext} \in \mathbb{R}^{\bar{n}}$ stands for the generalized external forces acting on the system.

To obtain a model representation suitable for a balancing controller, the joint coordinates of both feet are substituted by their Cartesian coordinates. The following task Jacobian $\mathbf{J} \in \mathbb{R}^{\bar{n} \times \bar{n}}$ provides the mapping into task space velocities $\dot{\mathbf{x}} \in \mathbb{R}^{\bar{n}}$:

$$\underbrace{\begin{pmatrix} \boldsymbol{\nu}_c \\ \boldsymbol{\nu} \\ \dot{\mathbf{q}}_f \end{pmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{Ad} & \mathbf{J}' \\ \mathbf{0} & \mathbf{S}_f \end{bmatrix}}_{\mathbf{J}} \begin{pmatrix} \boldsymbol{\nu}_c \\ \dot{\mathbf{q}} \end{pmatrix}, \quad (2)$$

where $\boldsymbol{\nu}_i = (\dot{\mathbf{x}}_i^T \boldsymbol{\omega}_i^T)^T \in \mathbb{R}^6$ with $i \in \{r, l\}$ is the translational and rotational velocity of the right and left foot, respectively, which are stacked together into the velocity vector $\boldsymbol{\nu} = (\boldsymbol{\nu}_r^T \boldsymbol{\nu}_l^T)^T$. The remaining free DoF $\dot{\mathbf{q}}_f \in \mathbb{R}^{n-12}$ are defined in joint space, with $\mathbf{S}_f \in \mathbb{R}^{(n-12) \times n}$ selecting the corresponding joints of the whole joint vector. The stacked adjoint matrices for the legs are denoted by $\mathbf{Ad} \in \mathbb{R}^{12 \times 6}$ and $\mathbf{J}' \in \mathbb{R}^{12 \times n}$ are the stacked Jacobians, respectively [22].

The CAM $\mathbf{l}_c \in \mathbb{R}^3$ depends linearly on the velocity vector

$$\mathbf{l}_c = \mathbf{A} \begin{pmatrix} \boldsymbol{\nu}_c \\ \dot{\mathbf{q}} \end{pmatrix} = \underbrace{\mathbf{A}\mathbf{J}^{-1}}_{\bar{\mathbf{A}}} \dot{\mathbf{x}}, \quad (3)$$

where $\mathbf{A} \in \mathbb{R}^{3 \times \bar{n}}$ is the rotational part of the centroidal momentum matrix (CMM) [23]. The CAM is represented in a frame attached to the CoM and aligned with the inertial frame. It can also be expressed as a function of the task velocities and a transformed CMM denoted by $\bar{\mathbf{A}}$ using inverse kinematics of (2). In this work, \mathbf{J} is a square matrix and assumed to be invertible; the handling of redundancy and singular configurations is considered as future work.

B. Passivity-based Whole-body Controller [20]–[22]

The controller generates ground reaction forces based on the robot's desired movement and compensates for possible disturbances. The derivation in this section assumes that the robot is balancing in double support; however, the formulation can be extended to further contact configurations by applying several modifications.

Inspired by PD+ control [26], the desired closed-loop behavior is formulated as follows:

$$\mathbf{M} \begin{pmatrix} \Delta \dot{\boldsymbol{\nu}}_c \\ \Delta \dot{\mathbf{q}} \end{pmatrix} + \mathbf{C} \begin{pmatrix} \Delta \boldsymbol{\nu}_c \\ \Delta \dot{\mathbf{q}} \end{pmatrix} = \boldsymbol{\tau}_{ext} - \mathbf{J}^T \begin{pmatrix} \mathbf{w}_c^{imp} \\ \mathbf{w}_{grf} \\ \boldsymbol{\tau}_f^{imp} \end{pmatrix}, \quad (4)$$

where $\mathbf{w}_{grf} \in \mathbb{R}^{12}$ denotes the concatenated contact wrenches of both feet and the deviation from the commanded trajectories are given by $\Delta \boldsymbol{\nu} = \boldsymbol{\nu}_c - \boldsymbol{\nu}_c^{cmd}$ and $\Delta \dot{\mathbf{q}} = \dot{\mathbf{q}} - \dot{\mathbf{q}}^{cmd}$. Note that the commanded trajectories are generated in task space and the corresponding commanded CoM and joint values are computed via inverse kinematics:

$$\begin{pmatrix} \boldsymbol{\nu}_c^{cmd} \\ \dot{\mathbf{q}}^{cmd} \end{pmatrix} = \mathbf{J}^{-1} \dot{\mathbf{x}}^{cmd}. \quad (5)$$

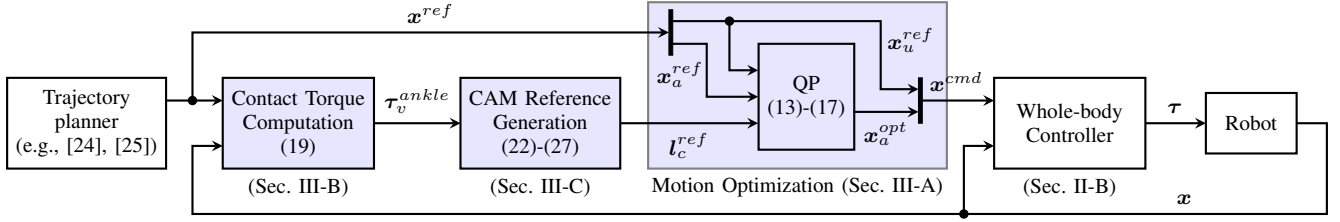


Fig. 2. Overview of the system architecture, where the blue highlighted elements are considered as the main contributions of this work. For brevity, time derivatives of system states and equations not used in the final implementation of the algorithm are omitted.

The CoM associated impedances are defined by

$$\mathbf{w}_c^{imp} = \begin{pmatrix} \mathbf{K}_c(\mathbf{x}_c - \mathbf{x}_c^{cmd}) + \mathbf{D}_c(\dot{\mathbf{x}}_c - \dot{\mathbf{x}}_c^{cmd}) \\ \boldsymbol{\Sigma}_b(\boldsymbol{\Sigma}_b, (\mathbf{R}_b^{cmd})^T \mathbf{R}_b) + \mathbf{B}_b(\boldsymbol{\omega}_b - \boldsymbol{\omega}_b^{cmd}) \end{pmatrix}, \quad (6)$$

where the linear and rotational stiffness matrices $\mathbf{K}_c > \mathbf{0}$ and $\boldsymbol{\Sigma}_b > \mathbf{0}$ as well as the linear and rotational damping matrices $\mathbf{D}_c > \mathbf{0}$ and $\mathbf{B}_b > \mathbf{0}$ are symmetric and positive definite. The Cartesian orientation of the hip is controlled by a virtual rotational spring $\boldsymbol{\tau}_r(\boldsymbol{\Sigma}_b, (\mathbf{R}_b^{cmd})^T \mathbf{R}_b)$ [20], while the impedance of the joint task is realized by

$$\boldsymbol{\tau}_f^{imp} = \mathbf{K}_f(\mathbf{q}_f - \mathbf{q}_f^{cmd}) + \mathbf{D}_f(\dot{\mathbf{q}}_f - \dot{\mathbf{q}}_f^{cmd}), \quad (7)$$

with the positive definite and symmetric linear spring $\mathbf{K}_f > \mathbf{0}$ and damper $\mathbf{D}_f > \mathbf{0}$ matrices.

By comparing the system dynamics (1) and the desired closed-loop behavior (4) while focusing on the upper six rows, we obtain the following equality

$$\underbrace{\mathbf{A}d^T \mathbf{w}_{grf}}_{\mathbf{w}_c^{ff}} = \mathbf{M}_c \begin{pmatrix} \dot{\mathbf{v}}_c^{cmd} \\ \ddot{\mathbf{q}}_c^{cmd} \end{pmatrix} + \mathbf{C}_c \begin{pmatrix} \mathbf{v}_c^{cmd} \\ \dot{\mathbf{q}}_c^{cmd} \end{pmatrix} - \mathbf{w}_g + \mathbf{w}_c^{imp}, \quad (8)$$

with the CoM-mapped contact wrench on the left-hand side and the overall desired CoM wrench on the right-hand side. Accordingly, the inertia and Coriolis matrix are split into the upper six rows describing the centroidal dynamics and the lower part, i.e., $\mathbf{M} = [\mathbf{M}_c^T, \mathbf{M}_q^T]^T$ and $\mathbf{C} = [\mathbf{C}_c^T, \mathbf{C}_q^T]^T$. The feedforward terms are summarized in \mathbf{w}_c^{ff} . Note that the transposed adjoint matrix, also referred to as contact map, has a full rank of six, while the size of \mathbf{w}_{grf} is 12 in double support. To determine the distribution of \mathbf{w}_{grf} in all possible configurations, a constrained QP based on (8) is formulated using the following contact constraints: contact unilaterality, Coulomb's friction model, bounded normal force, bounded torque on the z -axis, and CoP constraints. After resolving the wrench distribution, the final control torques are computed as follows:

$$\boldsymbol{\tau} = \mathbf{M}_q \begin{pmatrix} \dot{\mathbf{v}}_c^{cmd} \\ \ddot{\mathbf{q}}_c^{cmd} \end{pmatrix} + \mathbf{C}_q \begin{pmatrix} \mathbf{v}_c^{cmd} \\ \dot{\mathbf{q}}_c^{cmd} \end{pmatrix} - (\mathbf{J}')^T \mathbf{w}_{grf} - \mathbf{S}_f^T \boldsymbol{\tau}_f^{imp}. \quad (9)$$

III. CAM-BASED MOTION OPTIMIZATION

We propose a framework that takes as input whole-body trajectories from a high-level planner, e.g., [24], [25], and optimizes over the velocities of selected DoF w.r.t. a CAM reference which is generated online based on present external

disturbances. Note that a direct measurement of the external forces is not necessary.

A. Whole-body Motion Optimization

We separate the whole task space into k DoF adjusted within the motion optimizer and the remaining $\bar{n} - k$ DoF kept unchanged with $k \in \{0, \dots, \bar{n}\}$. The transformed CMM and the task space velocity vector in (3) can be accordingly split as follows:

$$\bar{\mathbf{A}} = \begin{bmatrix} \underbrace{\bar{\mathbf{A}}_1 \cdots \bar{\mathbf{A}}_k}_{\bar{\mathbf{A}}_a \in \mathbb{R}^{3 \times k}} & \underbrace{\bar{\mathbf{A}}_{k+1} \cdots \bar{\mathbf{A}}_{\bar{n}}}_{\bar{\mathbf{A}}_u \in \mathbb{R}^{3 \times (\bar{n}-k)}} \end{bmatrix}, \quad (10)$$

$$\dot{\mathbf{x}}^T = \begin{pmatrix} \underbrace{\dot{x}_1 \cdots \dot{x}_k}_{\dot{\mathbf{x}}_a^T \in \mathbb{R}^k} & \underbrace{\dot{x}_{k+1} \cdots \dot{x}_{\bar{n}}}_{\dot{\mathbf{x}}_u^T \in \mathbb{R}^{(\bar{n}-k)}} \end{pmatrix}, \quad (11)$$

where the subscriptions a and u denote adjusted and unchanged DoF, respectively. The optimization variables can be selected online depending on the contact configuration and the planning objective. For example, if only the velocities of the joint task should be adjusted w.r.t. a CAM objective, the task space velocity vector is assigned as follows: $\dot{\mathbf{x}}_a = \dot{\mathbf{q}}_f$ and $\dot{\mathbf{x}}_u = (\boldsymbol{\nu}_c^T \boldsymbol{\nu}^T)^T$.

The goal is to find optimized velocities $\dot{\mathbf{x}}_a^{opt}$ for the k selected DoF to induce a predefined reference CAM \mathbf{l}_c^{ref} to the system. As seen in Fig. 2, the reference CAM is obtained from the CAM reference generator, which will be presented in the Sections III-B and III-C. By reformulating (3), the optimized velocities are determined to satisfy

$$\mathbf{l}_c^{ref} = \bar{\mathbf{A}}_a \dot{\mathbf{x}}_a^{opt} + \bar{\mathbf{A}}_u \dot{\mathbf{x}}_u^{ref}, \quad (12)$$

where $\dot{\mathbf{x}}_u^{ref}$ is the reference velocity of the unchanged DoF provided by the trajectory planner. To solve (12) with linear inequality constraints, a QP is formulated

$$\min_{\dot{\mathbf{x}}_a^{opt}} \left(\frac{1}{2} \boldsymbol{\delta}_l^T \mathbf{Q}_l \boldsymbol{\delta}_l + \frac{1}{2} \boldsymbol{\delta}_p^T \mathbf{Q}_p \boldsymbol{\delta}_p \right), \quad (13)$$

with the residua

$$\boldsymbol{\delta}_l = \bar{\mathbf{A}}_a \dot{\mathbf{x}}_a^{opt} + \bar{\mathbf{A}}_u \dot{\mathbf{x}}_u^{ref} - \mathbf{l}_c^{ref}, \quad (14a)$$

$$\boldsymbol{\delta}_p = \dot{\mathbf{x}}_a^{opt} - \dot{\mathbf{x}}_a^d, \quad (14b)$$

and subject to the constraints

$$\dot{\mathbf{x}}_a^{min} \leq \dot{\mathbf{x}}_a^{opt} \leq \dot{\mathbf{x}}_a^{max}. \quad (15)$$

The desired velocity $\dot{\mathbf{x}}_a^d$ in (14b) is computed based on the deviation of the optimized posture from its reference

$$\dot{\mathbf{x}}_a^d = \dot{\mathbf{x}}_a^{ref} + \mathbf{K}_p(\mathbf{x}_a^{ref} - \mathbf{x}_a^{opt}), \quad (16)$$

where the reference velocity for the optimization variables $\dot{\mathbf{x}}_a^{ref}$ and the corresponding position \mathbf{x}_a^{ref} are provided by the trajectory planner. The optimized position \mathbf{x}_a^{opt} is obtained from $\dot{\mathbf{x}}_a^{opt}$ through integration w.r.t. time. The convergence behavior of the robot towards its reference pose can be influenced by the design parameter $\mathbf{K}_p > \mathbf{0}$ which is diagonal and positive definite.

The QP finds a trade-off between an angular momentum task (14a) and a reference posture task (14b). The non-strict priorities between the two tasks can be adjusted based on the choice of the weighting matrices \mathbf{Q}_l and $\mathbf{Q}_p > \mathbf{0}$, which are both positive definite and symmetric. A non-strict task hierarchy is necessary to allow the robot to fully converge to its reference posture even if $\mathbf{l}_c^{ref} \equiv \mathbf{0}$. The combination of an angular momentum and a posture task is chosen because optimizing only for the CAM (14a) may lead to unfavorable body postures. The reference posture provided by the trajectory planner is defined based on high-level task objectives.

The kinematic constraints in task space (15) ensure that predefined position limits ($\underline{\mathbf{x}}_a, \bar{\mathbf{x}}_a$) and velocity limits ($\underline{\dot{\mathbf{x}}}_a, \bar{\dot{\mathbf{x}}}_a$) are not exceeded

$$\begin{aligned} \dot{\mathbf{x}}_a^{min} &= \max(\mathbf{K}_a(\underline{\mathbf{x}}_a - \mathbf{x}_a^{opt}), \dot{\mathbf{x}}_a), \\ \dot{\mathbf{x}}_a^{max} &= \min(\mathbf{K}_a(\bar{\mathbf{x}}_a - \mathbf{x}_a^{opt}), \bar{\dot{\mathbf{x}}}_a), \end{aligned} \quad (17)$$

where $\mathbf{K}_a > \mathbf{0}$ is diagonal and positive definite. Larger values of \mathbf{K}_a lead to a wider area where the optimization (13) is uninfluenced; however, they also demand higher accelerations in the neighborhood of the constraint [27]. These constraints can be extended to a Cartesian-based self-collision avoidance or singular configurations prevention. Based on the optimized velocities, the corresponding commanded whole-body trajectories are generated through numerical differentiation and integration and finally concatenated with the unchanged DoF.

B. Hip and Ankle Strategy in the Context of Passivity-based Whole-body Control

For balancing scenarios, the high-level planner generates trajectories to hold a fixed posture, which means that the reference values on velocity and acceleration level are zero, $\dot{\mathbf{x}}^{ref} = \ddot{\mathbf{x}}^{ref} = \mathbf{0}$. Therefore, the whole-body motion resulting from an external push is solely generated through the methods presented in Section III.

The wrench distribution problem in Section II-B searches for feasible contact wrenches to produce the desired force and torque at the CoM, (8), counteracting disturbances. As the contact wrenches need to fulfill the contact constraints, an upper and lower limit of the maximum producible CoM wrench exists. Especially for the case disturbed by pushes, the contact torque constraints are easily reached, i.e., the CoP approaches the edge of the support area, and foot rotation or tilting can happen, which causes the robot to fall. To prevent this, an additional torque around the CoM is to be induced by actively generating CAM, which counteracts the present contact torques. In consequence, desired CoM wrenches of

higher magnitude can be generated, i.e., stronger pushes can be compensated.

The reference CAM is computed based on the required contact torques. To circumvent the wrench distribution problem and make the adjoint matrix in (8) invertible, the contact torques are calculated for a single virtual stance foot with its center located between the right and left foot position in double support, i.e., $\mathbf{x}_v = \frac{1}{2}(\mathbf{x}_r + \mathbf{x}_l)$, or equal to the foot position of the stance foot in single support. The virtual stance foot has a size equal to the robot's support area. By applying this simplification and reformulating (8), the contact torque of the virtual stance foot $\boldsymbol{\tau}_v \in \mathbb{R}^3$ is obtained as

$$\boldsymbol{\tau}_v = \underbrace{[\hat{\mathbf{x}}_{v,c} \mathbf{I}]}_{\mathbf{Ad}_{v,rot}^{-T}} (\mathbf{w}_c^{ff} - \mathbf{w}_g + \mathbf{w}_c^{imp}), \quad (18)$$

where $\hat{\mathbf{x}}_{v,c}$ is the cross-product matrix of the vector $\mathbf{x}_{v,c} = \mathbf{x}_c - \mathbf{x}_v$ between the CoM and the virtual stance foot position. The rotational part of the inverted adjoint matrix of the virtual stance foot, which has full rank by construction, is represented by $\mathbf{Ad}_{v,rot}^{-T}$. The contact torques in (18) can be further split as follows:

$$\boldsymbol{\tau}_v = \underbrace{\mathbf{Ad}_{v,rot}^{-T} \mathbf{w}_c^{ff}}_{\boldsymbol{\tau}_v^{hip}} + \underbrace{\mathbf{Ad}_{v,rot}^{-T} (-\mathbf{w}_g + \mathbf{w}_c^{imp})}_{\boldsymbol{\tau}_v^{ankle}}, \quad (19)$$

where $\boldsymbol{\tau}_v^{hip}$ represents the contact torques generated by intended upper body motion, here in the form of the feedforward terms, and is therefore associated with a hip strategy. The contact torques induced by gravity and impedance terms are summarized by $\boldsymbol{\tau}_v^{ankle}$, which are considered as an ankle strategy since they indicate the resulting contact torque that would be present if a fixed body posture is preserved, i.e., if $\mathbf{w}_c^{ff} = \mathbf{0}$.

The feedforward terms in (8) can be reformulated with expanded inertia and Coriolis matrix as follows:

$$\underbrace{\begin{pmatrix} \mathbf{f}_c^{ff} \\ \boldsymbol{\tau}_b^{ff} \end{pmatrix}}_{\mathbf{w}_c^{ff}} = \underbrace{\begin{bmatrix} m\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{\omega\omega} & \mathbf{M}_{\omega q} \end{bmatrix}}_{\mathbf{M}_c} \begin{pmatrix} \ddot{\mathbf{x}}_c^{cmd} \\ \dot{\boldsymbol{\omega}}_b^{cmd} \\ \ddot{\mathbf{q}}^{cmd} \end{pmatrix} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{C}_\omega \end{bmatrix}}_{\mathbf{C}_c} \begin{pmatrix} \dot{\mathbf{x}}_c^{cmd} \\ \boldsymbol{\omega}_b^{cmd} \\ \dot{\mathbf{q}}^{cmd} \end{pmatrix}, \quad (20)$$

where the feedforward force and torque are \mathbf{f}_c^{ff} and $\boldsymbol{\tau}_b^{ff}$, respectively. The feedforward torque can be interpreted as the rate of change of CAM that results from the commanded trajectory, which is transformed via inverse kinematics (5) from task space. The commanded CoM velocity and acceleration, and therefore also \mathbf{f}_c^{ff} , is defined to be zero for balancing scenarios. Consequently, the contact hip torque simplifies to

$$\boldsymbol{\tau}_v^{hip} = \boldsymbol{\tau}_b^{ff} = \dot{\mathbf{i}}_c^{cmd}, \quad (21)$$

where $\dot{\mathbf{i}}_c^{cmd}$ is the rate of change of the commanded CAM. Note that the commanded CAM is the output of the motion optimization in Section III-A, whereas the reference CAM is its input.

C. CAM Reference Generation for Force-disturbed Balancing Scenarios

In this section, a method is presented for scheduling the hip strategy's activation time, and a corresponding CAM reference is provided. Based on τ_v^{ankle} , (19), a reference CAM and therefore a resulting τ_v^{hip} will be generated to keep τ_v within its limits $\tau_v^{min/max} \in \mathbb{R}^3$. The limits on the torque around the z -axis are approximated by predefined upper and lower values [21], [22]. The limits for the xy -plane are computed based on the constraint that the combined CoP $\mathbf{p}_v \in \mathbb{R}^2$ needs to be inside the virtual stance foot \mathcal{S}_v

$$\mathbf{p}_v = \frac{1}{f_{v,z}} \begin{pmatrix} -\tau_{v,y} \\ \tau_{v,x} \end{pmatrix} \in \mathcal{S}_v, \quad (22)$$

where $f_{v,z}$ is the total vertical force applied to the ground.

To account for the limited angular momentum storing capabilities of a humanoid, the CAM reference generation is separated into three consecutive phases: 1) the first phase actively generates CAM to compensate for the external forces evoked by a disturbance (e.g. push); 2) the second phase brings the CAM reference back to zero to bring the robot to a halt; and 3) the last phase ensures that the robot converges to its reference pose again. A switch back to phase 1 is possible at any time if an additional push is detected. The push recovery algorithm is applied independently for every spatial dimension $i \in \{x, y, z\}$ aligned with the (potentially rotated) virtual stance foot, i.e., pushes from all directions can be independently compensated in the respective axis. For the sake of clarity, the indices i are omitted from here on. The presented relations in scalar form apply to each spatial dimension. The proposed algorithm is explained with simulation results using the humanoid robotics platform OpenHRP [28] with the same robot model employed in experimental verifications (Section IV).

1) *CAM Generation Phase:* The first phase is activated after a push occurs, which is detected when the corresponding τ_v^{ankle} exceeds a predefined threshold τ_v^{thres} . The threshold is a function of the contact torque limits resulting from (22)

$$\tau_v^{thres} = \alpha \tau_v^{min/max}, \quad (23)$$

with the design parameter $\alpha \in [0, 1]$. The larger α , the later the hip strategy will be activated. In general, the hip strategy should be activated as late as possible to fully exploit the capabilities of the ankle strategy first. However, if the hip strategy only gets active when the contact constraints are reached ($\alpha = 1$), small model uncertainties or tracking errors can cause the robot to fall. Therefore, a safety margin improves the robustness of the algorithm; in practice, we achieved good results with $\alpha \in [0.7, 0.9]$. Fig. 3a shows a simulated course of τ_v^{ankle} from (19). The robot is pushed from behind with an impulse of 12 Ns, which, normalized by its mass, corresponds to a delta velocity of $0.152 \frac{m}{s}$.

The rate of change of the reference CAM is defined as the difference of the ankle torque and its threshold

$$\dot{l}_c^{ref}(t) = \tau_v^{thres}(t) - \tau_v^{ankle}(t). \quad (24)$$

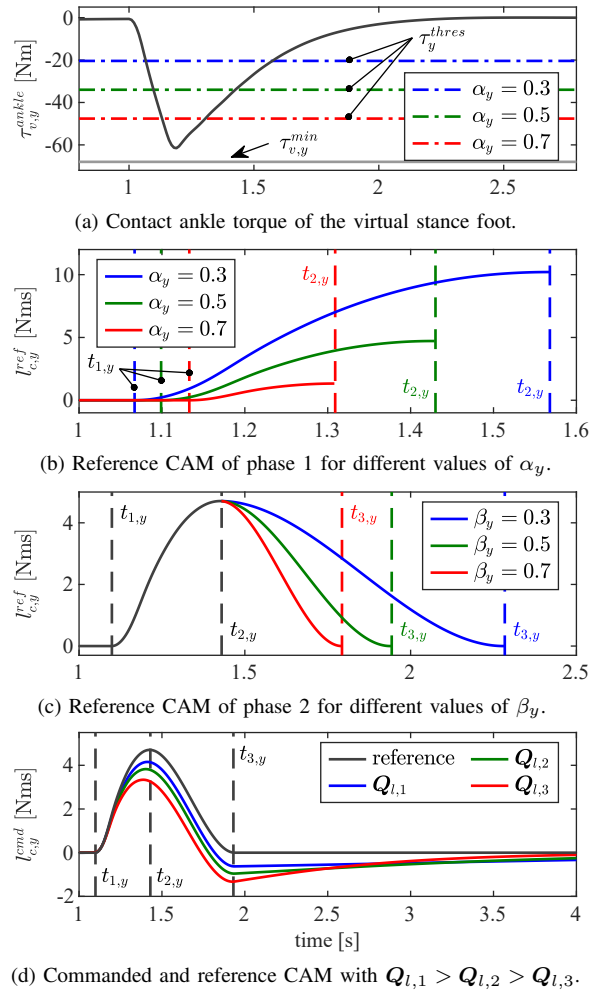


Fig. 3. CAM reference generation in three phases based on the contact ankle torque resulting from a push in positive x -direction applied at $t = 1$ s. The vertical dashed lines denote the beginning of the respective phase.

This difference can be interpreted as the additional hip torque in (19) that is needed to keep τ_v within the limits. The corresponding reference CAM $l_c^{ref}(t)$ can be obtained by integrating (24) with respect to time. The time t is defined within the interval $t \in [t_1, t_2)$, where the moment of activation of the first phase is characterized by $t = t_1$. Phase 1 is completed when the ankle torque is again below the predefined threshold, i.e., $|\tau_v^{ankle}(t)| < \tau_v^{thres}(t)$, this point in time is defined by $t = t_2$. Fig. 3b shows the resulting CAM reference for different values of α .

2) *CAM Reduction Phase:* During phase 1, the reference CAM is monotonically increasing. After the push is compensated, the reference CAM has a non-zero value, which needs to be smoothly reduced to zero to bring the robot to rest again. Therefore, a third-order polynomial is used to generate a trajectory while ensuring C^1 continuity to avoid jumps in the resulting τ_v^{hip} .

The system state when phase 2 gets activated is used as an initial condition for the polynomial. In order to bring the robot to a halt after phase 2, the CAM and its time derivative need to be zero at the end of the trajectory. Phase 2 is defined within the time interval $t \in [t_2, t_3]$. The CAM reference

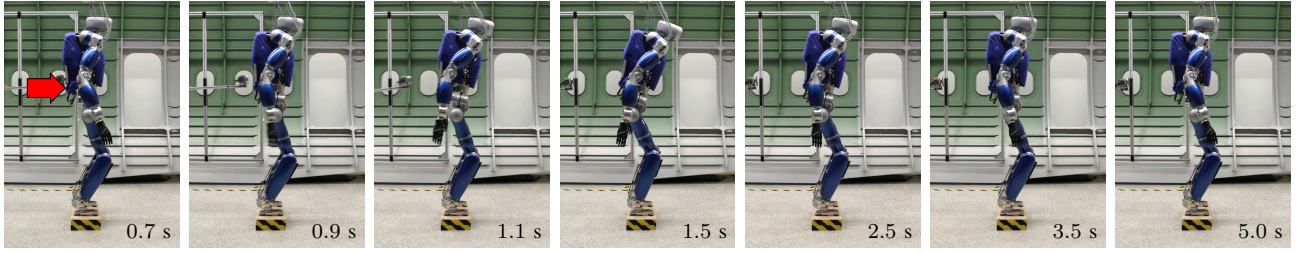


Fig. 4. Snapshots of experiment #1: Push recovery in double support. The red arrow indicates the direction of the push.

trajectory is formulated as

$$l_c^{ref}(t) = \sum_{j=0}^3 a_j \frac{(t-t_2)^j}{T^j}, \quad (25)$$

where the coefficients a_j can be determined based on the boundary conditions. The last free parameter is the total duration of phase 2 which is denoted by $T = t_3 - t_2$. Based on the critical points of l_c^{ref} , a relationship between the total duration, the reference CAM at the beginning of phase 2 l_c^{ref} , and its maximum rate of change $\dot{l}_c^{max} > 0$ is obtained as follows:

$$T = \frac{3 |l_c^{ref}|}{2 \dot{l}_c^{max}}. \quad (26)$$

Note that T cannot be chosen arbitrary large without reaching kinematic limits, while small T induce a high maximum rate of change of CAM. To find a trade-off, \dot{l}_c^{max} is chosen as a function of the peak rate of change of CAM during phase 1 given by

$$\dot{l}_c^{max} = \beta \left(\max_{t_1 \leq t < t_2} |\dot{l}_c^{ref}(t)| \right), \quad (27)$$

with the design parameter defined within the interval $\beta \in (0, 1]$, see Fig. 3c. Small values of β increase the risk of activating position limits while large values can lead to an aggressive reduction of the CAM. Both scenarios can deflect the CoP towards the edge of the support area and potentially cause the robot to fall over. We achieved good results with $\beta \in [0.3, 0.5]$. Note, this formulation gives a good intuition on how fast the CAM should be reduced but does not provide a formal guarantee that kinematic limits or contact constraints will not be reached.

3) *Posture Recovery Phase*: After phase 2 is completed, the reference CAM is reduced to zero, but the robot configuration is still deviating from its reference pose. At this point, the posture task (14b) of the motion optimization becomes dominant. It provides a smooth return to the initial robot configuration while inducing only a small CAM with an opposite sign to the one generated in phases 1 and 2. The CAM tracking depends on the weight matrix selection. Increasing Q_l in (13) while keeping Q_p constant leads to better CAM tracking but also increases the convergence time w.r.t. the reference pose and consequently the total duration of phase 3, see Fig. 3d.

IV. EXPERIMENTS

The proposed methods were validated in experiments with the torque-controlled humanoid robot TORO [29]. The robot

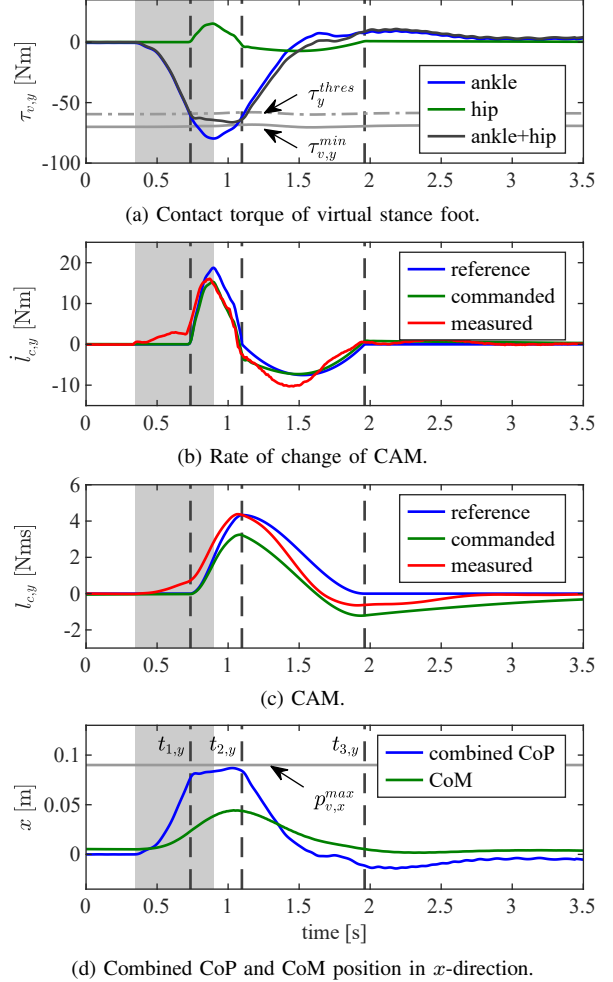


Fig. 5. Results of the experiment #1: Push recovery in double support, where the gray highlighted area indicates the duration of the push and the vertical dashed lines denote the beginning of the respective phase.

has 27 DoF, a height of 1.74 m, and a total weight of 79.2 kg. The pushing force is measured by an external force sensor mounted on the tip of a stick. Footage of the experiments can be found in the supplementary video.

A. Implementation Details

The constrained QPs of the wrench distribution in Section II-B and the motion optimization (13) are solved by using two independent instances of the online active-set optimization algorithm qpOASES [30], which provides a ‘hot-start’ functionality; good results were achieved with a maximum of 10 working set recalculations. The controller

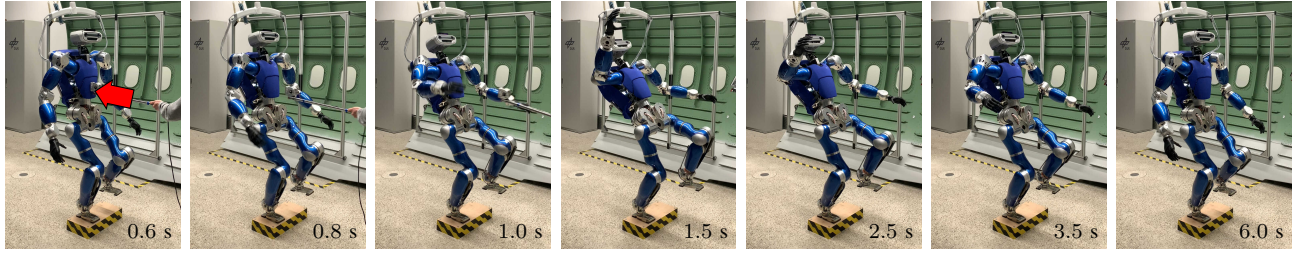


Fig. 6. Snapshots of experiment #2: Push recovery in single support. The red arrow indicates the direction of the push.

is executed at a speed of 1 kHz, whereas the low-level joint controller computation rate is 3 kHz [31].

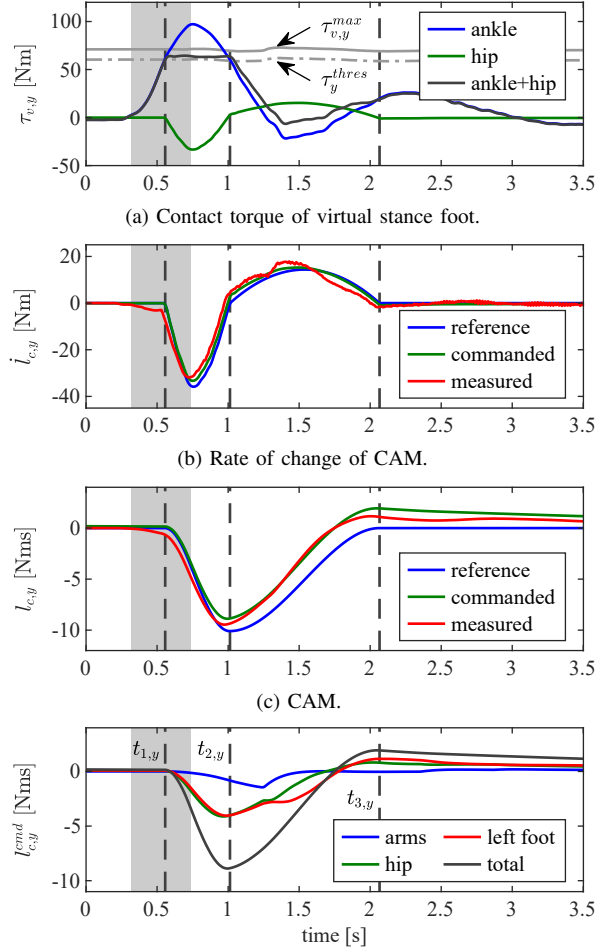
The activation threshold for phase 1 is defined by $\alpha_y = 0.9$; the duration of phase 2 is set by $\beta_y = 0.4$. Due to high sensor noise, the measured CAM l_c and its rate of change \dot{l}_c are reproduced by using a momentum-based disturbance observer similar to the one in [27]. Note that these values are only for evaluation and not used within the presented algorithms.

B. Experimental Evaluations

We present two scenarios to evaluate the push recovery algorithm's performance on the real system. In the first scenario, the robot is standing upright in double support and is pushed impulsively from behind (along the x -axis) on the height of the hip with a maximum force of 65 N and an impulse of 18 Ns ($0.227 \frac{m}{s}$). The optimization variable \hat{x}_a^{opt} in (13) includes the angular velocities of the hip around all three axes, the joint velocity in the torso as well as the first four joint velocities in every arm. To reject the push, the robot generates CAM by bending forward and moving the arms back. Fig. 4 shows snapshots of the robot behavior as a reaction to the push.

The ankle, hip, and resulting contact torque of the virtual stance foot are shown in Fig. 5a. The pure ankle torque exceeds its limit, which means that the combined CoP would have reached the support area's edge if a fixed posture had been preserved, causing the robot to fall. The controller from Section II-B alone could not have compensated the push without the presented CAM-based motion optimization from Section III. Through the additional generated CAM and corresponding τ_v^{hip} , the resulting contact torque τ_v is kept within its limits and the robot recovers its balance after the push. Fig. 5d shows the corresponding combined CoP and CoM deviation. The course of the reference, commanded and measured CAM through all phases and their rate of change are plotted in Figs. 5b and 5c, respectively.

In the second experiment, the robot balances on the right leg and is pushed from the front with a maximum force of 75 N and an impulse of 22.5 Ns ($0.284 \frac{m}{s}$). Snapshots of the experiment are presented in Fig. 6, while the results are shown in Fig. 7. Note that the translational Cartesian velocity of the left foot is added to the optimization variables of the previous experiment with the double support; the corresponding CoM-mapped impedance wrench needs to be added in (8) and in (18) to the ankle torque (refer to [22] for further details). Exemplary for the DoF adjusted in the



(d) Contribution of different body segments to commanded CAM.

Fig. 7. Results of the experiment #2: Push recovery in single support, where the gray highlighted area indicates the duration of the push and the vertical dashed lines denote the beginning of the respective phase.

motion optimization in (13), the course of the hip angular velocity around the pitch axis, the Cartesian velocity of the swing leg in the x -direction, and the velocity of the first shoulder joint with the corresponding positions and limits are displayed in Fig. 8.

Note that the robot can compensate pushes in the sagittal plane of higher magnitude in single support compared to the same test setup in double support. The extent of the support area in x -direction is equal for both configurations; however, in single support, the swing leg generates almost one-half of the peak CAM in the y -direction, see Fig. 7d.

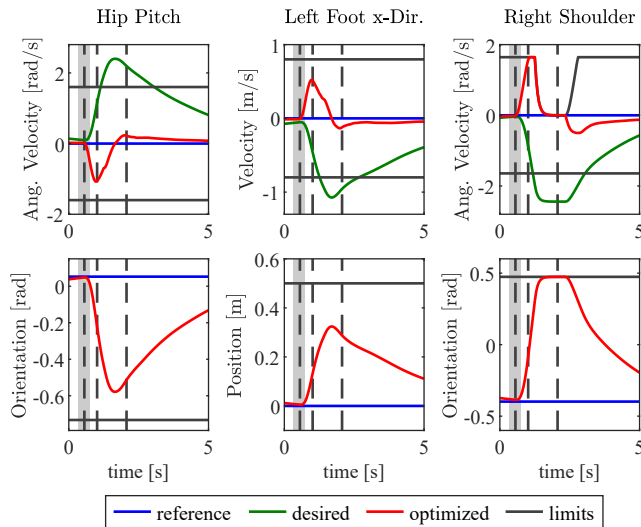


Fig. 8. A selection of three optimized velocities and positions in the motion optimizer with limits (17), where the gray area indicates the pushing and the dashed lines denote the beginning of the respective phase.

V. CONCLUSION

This paper proposes a push recovery algorithm for balancing scenarios exploiting an ankle and hip strategy. When a push occurs, the activation of an additional hip strategy is scheduled, and a corresponding reference CAM is generated based on the induced contact torque. The subsequent motion optimizer finds a trade-off between an angular momentum and a posture task to generate feasible whole-body trajectories. The proposed approach was validated in experiments for single and double support balancing scenarios. It was shown that the CAM-based motion optimization enables the robot to maintain balance against pushes, which could not be handled by the controller of Section II-B alone, while kinematic limits were additionally incorporated. The hip strategy can be activated independently of task conflicts or reached contact constraints, which facilitates the incorporation of model uncertainties or tracking errors and increases the robustness.

As part of future work, methods for automatically tuning the design parameters and an improved representation of the torque limits on the z -axis will be evaluated.

REFERENCES

- [1] L. M. Nashner and G. McCollum, "The organization of human postural movements: A formal basis and experimental synthesis," *Behav. Brain Sciences*, vol. 8, no. 1, p. 135–150, 1985.
- [2] B. E. Maki and W. E. McIlroy, "The role of limb movements in maintaining upright stance: the "change-in-support" strategy," *Physical Therapy*, vol. 77, pp. 488–507, Jun. 1997.
- [3] C. F. Runge, C. L. Shupert, F. B. Horak, and F. E. Zajac, "Ankle and hip postural strategies defined by joint torques," *Gait Posture*, vol. 10, no. 2, pp. 161–170, Oct. 1999.
- [4] M. Popovic, A. Hofmann, and H. Herr, "Angular momentum regulation during human walking: biomechanics and control," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 3, 2004, pp. 2405–2411.
- [5] H. Herr and M. Popovic, "Angular momentum in human walking," *J. Exp. Biol.*, vol. 211, no. 4, pp. 467–481, Feb. 2008.
- [6] S. Kajita *et al.*, "Resolved momentum control: humanoid motion planning based on the linear and angular momentum," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 2, Nov. 2003, pp. 1644–1650.

- [7] C. G. Atkeson and B. J. Stephens, "Multiple balance strategies from one optimization criterion," in *Proc. 7th IEEE-RAS Int. Conf. Humanoid Robots*, 2007, pp. 57–64.
- [8] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *Proc. 6th IEEE-RAS Int. Conf. Humanoid Robots*, Jan. 2006, pp. 200–207.
- [9] B. J. Stephens, "Humanoid push recovery," in *Proc. 7th IEEE-RAS Int. Conf. Humanoid Robots*, 2007, pp. 589–595.
- [10] B. J. Stephens and C. G. Atkeson, "Dynamic balance force control for compliant humanoid robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 1248–1255.
- [11] G. Wiedebach *et al.*, "Walking on partial footholds including line contacts with the humanoid robot atlas," in *Proc. 16th IEEE-RAS Int. Conf. Humanoid Robots*, 2016, pp. 1312–1319.
- [12] R. Hinata and D. N. Nenchev, "Balance stabilization with angular momentum damping derived from the reaction null-space," *Proc. 18th IEEE-RAS Int. Conf. Humanoid Robots*, pp. 188–195, 2018.
- [13] S. Hyon, R. Osu, and Y. Otaka, "Integration of multi-level postural balancing on humanoid robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Jun. 2009, pp. 1549–1556.
- [14] A. Hofmann, M. Popovic, and H. Herr, "Exploiting angular momentum to enhance bipedal center-of-mass control," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 4423–4429.
- [15] S.-H. Lee and A. Goswami, "A momentum-based balance controller for humanoid robots on non-level and non-stationary ground," *Auton. Robots*, vol. 33, pp. 399–414, Nov. 2012.
- [16] A. Herzog *et al.*, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Auton. Robots*, vol. 40, no. 3, pp. 473–491, 2016.
- [17] M. de Lasa, I. Mordatch, and A. Hertzmann, "Feature-based locomotion controllers," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 131:1–10, Jul. 2010.
- [18] M. Al Borno, E. Fiume, A. Hertzmann, and M. de Lasa, "Feedback control for rotational movements in feature space," *Comput. Graph. Forum*, vol. 33, no. 2, p. 225–233, May. 2014.
- [19] P. M. Wensing and D. E. Orin, "Improved computation of the humanoid centroidal dynamics and application for whole-body control," *Int. J. Humanoid Robotics*, vol. 13, no. 1, pp. 1550039:1–23, 2016.
- [20] C. Ott, M. A. Roa, and G. Hirzinger, "Posture and balance control for biped robots based on contact force optimization," in *Proc. 11th IEEE-RAS Int. Conf. Humanoid Robots*, 2011, pp. 26–33.
- [21] B. Henze, M. A. Roa, and C. Ott, "Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios," *Int. J. Robot. Res.*, vol. 35, no. 12, pp. 1522–1543, 2016.
- [22] G. Mesesan *et al.*, "Dynamic walking on compliant and uneven terrain using DCM and passivity-based whole-body control," in *Proc. 19th IEEE-RAS Int. Conf. Humanoid Robots*, 2019, pp. 25–32.
- [23] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Auton. Robots*, vol. 35, no. 2–3, pp. 161–176, Oct. 2013.
- [24] J. Engelsberger, G. Mesesan, and C. Ott, "Smooth trajectory generation and push-recovery based on divergent component of motion," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2017, pp. 4560–4567.
- [25] G. Mesesan, J. Engelsberger, C. Ott, and A. Albu-Schäffer, "Convex properties of center-of-mass trajectories for locomotion based on divergent component of motion," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3449–3456, 2018.
- [26] B. Paden and R. Panja, "Globally asymptotically stable 'PD+' controller for robot manipulators," *Int. J. Control*, vol. 47, no. 6, pp. 1697–1712, 1988.
- [27] J. Engelsberger, "Combining reduced dynamics models and whole-body control for agile humanoid locomotion," Ph.D. dissertation, Tech. Univ. Munich, Munich, Dec. 2016.
- [28] F. Kanehiro, H. Hirukawa, and S. Kajita, "OpenHRP: Open architecture humanoid robotics platform," *Int. J. Robot. Res.*, vol. 23, no. 2, pp. 155–165, Feb. 2004.
- [29] J. Engelsberger *et al.*, "Overview of the torque-controlled humanoid robot TORO," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2014, pp. 916–923.
- [30] H. J. Ferreau, H. G. Bock, and M. Diehl, "An online active set strategy to overcome the limitations of explicit MPC," *Int. J. Robust Nonlinear Control*, vol. 18, no. 8, pp. 816–830, 2008.
- [31] A. Albu-Schäffer *et al.*, "The DLR lightweight robot: design and control concepts for robots in human environments," *Ind. Robot: Int. J.*, vol. 34, no. 5, pp. 376–385, 2007.