Online Learning of Centroidal Angular Momentum Towards Enhancing DCM-Based Locomotion

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

Online Learning of Centroidal Angular Momentum Towards Enhancing DCM-Based Locomotion

Robert Schuller, George Mesesan, Johannes Englsberger, Jinoh Lee, and Christian Ott

***Abstract*— Gait generation frameworks for humanoid robots typically assume a constant centroidal angular momentum (CAM) throughout the walking cycle, which induces undesirable contact torques in the feet and results in performance degradation. In this work, we present a novel algorithm to learn the CAM online and include the obtained knowledge within the closed-form solutions of the Divergent Component of Motion (DCM) locomotion framework. To ensure a reduction of the contact torques at the desired center of pressure position, a CAM trajectory is generated and explicitly tracked by a whole-body controller. Experiments with the humanoid robot TORO demonstrate that the proposed method substantially increases the maximum step length and walking speed during locomotion.**

I. Introduction

Legged locomotion of robotic systems is a complex problem due to the nonlinear and underactuated characteristics of the corresponding multibody dynamics and the constraints on applicable contact forces. To handle the complexity, reduced models focusing on the center of mass (CoM) dynamics were introduced, one prominent example being the Linear Inverted Pendulum (LIP) model [1]. Based on the LIP model and the Zero-Moment Point (ZMP) [2], several methods for generating CoM trajectories were developed, e.g., [3], [4]. The Divergent Component of Motion (DCM) [5], [6], also referred to as instantaneous Capture Point [7], was introduced to simplify three-dimensional gait generation by focusing on the unstable part of the CoM dynamics. Based on the DCM concept, closed-form CoM trajectories can be generated efficiently using matrix-vector computation [8]–[10].

When using the DCM framework for gait generation, it is usually assumed that the rate of change of the centroidal angular momentum (CAM) is zero [5]. Consequently, the enhanced Centroidal Momentum Pivot (eCMP) is chosen to coincide with the desired center of pressure (CoP) position, as shown in Fig. 1. However, this assumption does not hold for the multibody dynamics of a humanoid robot, where a substantial CAM is generated through swing leg and pelvis motions. The induced CAM, if not adequately controlled, leads to a deviation of the actual CoP from its desired position. If the CoP approaches the edge of the support area, foot tilting can occur, which may cause the robot to fall over.

The inclusion of the multibody CAM in online gait generation is still an ongoing research problem due to its nonlinear and nonholonomic properties. In [11], a centroidal momentum trajectory, including linear and angular momenta, is optimized based on a kino-dynamic planning procedure. Nonetheless, this method is computationally heavy and performed offline. Other approaches provide extensions of the LIP model to give a better approximation of the multibody CAM [12]–[14]. In [15], the CAM is approximated by a simplified three-mass model and integrated within the DCM planning framework of [5]. However, these approximations still cannot cover the entire complexity of the CAM of a humanoid.

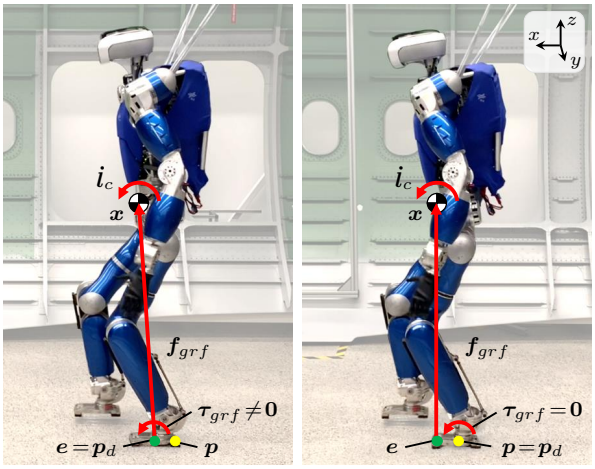


Fig. 1. The left diagram depicts the humanoid robot TORO walking with constant CAM assumption, while the right diagram shows it walking with the proposed method of online learning of CAM. The following quantities are presented: CoM ($\boldsymbol{x}$), eCMP ($\boldsymbol{e}$), CoP ($\boldsymbol{p}$), rate of change of CAM ($\dot{\boldsymbol{l}}_c$), external force vector ($\boldsymbol{f}_{grf}$), and contact torque ($\boldsymbol{\tau}_{grf}$).

Other methods do not explicitly account for the rotational dynamics during planning but compensate for unmodeled effects which result from the discrepancy between the LIP and the multibody model. A common approach is to use a preview controller in combination with a dynamic filter [16]. An expected ZMP error is computed for a given preview horizon from the multibody dynamics, and a second stage preview controller is applied to account for the deviation. This method is extended in [17] to adapt online the CoM and ZMP trajectories within a shorter reaction time. By exploiting the periodicity of walking motions, ZMP tracking can also be ensured by applying a model-free iterative learning controller (ILC) in combination with a preview controller [18].

More recently, the authors in [19] presented a DCM-based ILC approach to increase the robustness of the walking motions through an online adapted Virtual Repellent Point

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. Email: robert.schuller@dlr.de

(VRP) trajectory. This method improves the VRP tracking, but since the desired VRP trajectory is planned with a constant CAM assumption, the actual CoP is still deviating from its desired position during walking.

The presented methods [15]–[18] focus only on the ZMP tracking, which corresponds to a reduction of the horizontal (xy-direction) contact torques in the desired ZMP position for flat-ground walking [20]. Moreover, another highly limiting factor for robust and dynamic walking is the contact torque constraints about the vertical axis w.r.t. the contact surface (z-axis). Substantial torques about the z-axis can result in rotational slippage that destabilizes the walking. The approaches in [21]–[23] tackle this problem by canceling out the CAM about the z-axis; however, these methods do not provide explicit ZMP tracking.

In this work, we present a unified solution that ensures a contact torque reduction at the desired CoP position about all three axes by considering the multibody CAM during DCM-based gait generation. Similar to [15], we express the CAM as a polynomial function, but instead of using a simplified approximation, the CAM trajectory is learned online by exploiting the cyclic walking motions. After each walking phase, a polynomial function is fit online to the learned CAM, and the corresponding polynomial coefficients are used within the next walking phase to replan the CoM trajectory. Additionally, we provide an online updating scheme for the DCM trajectory to ensure continuity. Complementing the CoM reference, we generate a desired CAM trajectory, which is tracked by the whole-body controller introduced in [24] with an additional CAM-based motion optimization [25]. Fig. 2 overviews the proposed system architecture.

II. Fundamentals

This section provides the fundamentals of the DCM framework [5], [8]–[10]. Moreover, insights from our previous work [25] about the CAM in humanoid locomotion and its relation to the DCM framework are provided.

A. DCM Framework

The DCM $\boldsymbol{\xi} \in \mathbb{R}^3$ is defined as

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

with the CoM position $\boldsymbol{x} \in \mathbb{R}^3$ and velocity $\dot{\boldsymbol{x}} \in \mathbb{R}^3$. The DCM time constant is given by $b = \sqrt{\frac{\Delta z}{g}}$, where Δz is the nominal CoM height above the ground surface, and g is the gravitational constant. The unstable first-order dynamics of the DCM is given by

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

where $\boldsymbol{v} \in \mathbb{R}^3$ is the VRP which encodes the effects of external force and gravity. The VRP itself is located above the eCMP $\boldsymbol{e} \in \mathbb{R}^3$, with a constant height offset corresponding to the gravitational force, i.e., $\boldsymbol{v} = \boldsymbol{e} + (0,\ 0,\ \Delta z)^T$. The eCMP itself encodes the external forces[1] $\boldsymbol{f}_{grf} \in \mathbb{R}^3$ via

$$\boldsymbol{f}_{grf} = \frac{m}{b^2}(\boldsymbol{x} - \boldsymbol{e}) \tag{3}$$

[1]Here, we only consider ground reaction forces as external forces.

with m being the total mass of the robot. The corresponding Centroidal Momentum Pivot (CMP) [26] is defined as the point where a line connecting the CoM and eCMP passes through the ground surface.

B. DCM Reference Trajectory Generation

The overall motion is split into a sequence of n_φ transition phases, e.g., single support or double support phases, for which reference trajectories are computed. We provide closed-form DCM trajectory solutions for each transition phase based on an arbitrary polynomial eCMP reference trajectory given as

$$\boldsymbol{e}(t) = \sum_{j=0}^{n_v} \boldsymbol{a}_{v,j} t^j. \tag{4}$$

Here, n_v is the degree and $\boldsymbol{a}_{v,j} \in \mathbb{R}^3$ are the coefficients of the polynomial function representing the eCMP trajectory. The local time of the transition phase φ is given by $t \in [0,\ T_\varphi]$, with T_φ denoting the phase duration. By grouping the coefficients of (4) and considering the constant vertical offset to the eCMP, the VRP reference trajectory can be expressed as a polynomial of the following form:

$$\boldsymbol{v}(t) = \boldsymbol{P}_v\, \boldsymbol{t}(t). \tag{5}$$

The time is vectorized to $\boldsymbol{t}(t) = (1, t, t^2, ..., t^{n_v})^T$ and the polynomial parameter matrix is denoted by $\boldsymbol{P}_v \in \mathbb{R}^{3\times(n_v+1)}$.

Inserting (5) into (2) and solving the differential equation using the mathematical insights from [10], we get

$$\boldsymbol{\xi}(t) = e^{\frac{t-T_\varphi}{b}}\boldsymbol{\xi}_T + \boldsymbol{P}_v\boldsymbol{C}^T\Big(\boldsymbol{t}(t) - e^{\frac{t-T_\varphi}{b}}\boldsymbol{t}(T_\varphi)\Big), \tag{6}$$

where $\boldsymbol{\xi}_T$ is the given terminal DCM. The coefficient matrix $\boldsymbol{C} \in \mathbb{R}^{(n_v+1)\times(n_v+1)}$ is solely a function of the parameter b. The corresponding CoM trajectory can be obtained by solving (1) after inserting (6). For further details please refer to [10].

We obtain a continuous multi-step preview for the closed-form CoM trajectory by linking the single transition phases together. To bring the robot to a halt at the end of the trajectory, the terminal DCM of the last transition phase is chosen to coincide with the final VRP position [8], [9].

C. Multibody Dynamics and Centroidal Angular Momentum

The CAM $\boldsymbol{l}_c \in \mathbb{R}^3$ of a system depends linearly on the task space velocity vector

$$\boldsymbol{l}_c = \bar{\boldsymbol{A}} \begin{pmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{\chi}} \end{pmatrix}, \tag{7}$$

where $\bar{\boldsymbol{A}} \in \mathbb{R}^{3\times n}$ is the rotational part of the centroidal momentum matrix (CMM) [27] transformed into the n-dimensional task space. The task velocities besides the CoM velocity are represented by $\dot{\boldsymbol{\chi}} \in \mathbb{R}^{n-3}$. One possible choice for $\dot{\boldsymbol{\chi}}$ may include the angular velocity of the base, the feet velocities in Cartesian space, as well as the joint velocities of the upper body. The rate of change of the CAM corresponds to the total torque acting about the CoM, i.e. $\dot{\boldsymbol{l}}_c = \boldsymbol{\tau}_c$. The

contact torque at the desired CoP position $\boldsymbol{p}_d \in \mathbb{R}^3$ is given by

$$\boldsymbol{\tau}_{grf} = (\boldsymbol{x} - \boldsymbol{p}_d) \times \boldsymbol{f}_{grf} + \dot{\boldsymbol{l}}_c. \tag{8}$$

Consequently, the offset between the actual and desired CoP is a function of the contact torque

$$\boldsymbol{p} - \boldsymbol{p}_d = \frac{1}{m(g+\ddot{z})} \begin{pmatrix} \tau_{grf,y} \\ -\tau_{grf,x} \\ 0 \end{pmatrix}. \tag{9}$$

Here, $\ddot{z}$ represents the vertical CoM acceleration.

III. Problem Statement

To ensure that the desired and actual CoP positions in (9) coincide, the contact torque at the desired CoP position needs to be zero. However, if constant CAM is assumed during planning, i.e., $\boldsymbol{\tau}_c = \boldsymbol{0}$, and the eCMP is chosen to coincide with the desired CoP, the cross product on the right-hand side of (8) vanishes. This can be shown by inserting the external force vector from (3) into (8). In this case, the entire rate of change of CAM, which is induced through swing leg motion, should be regulated to be zero by a whole-body controller to ensure zero contact torque. However, this typically leads to extensive and inefficient upper-body motions.

Our goal is to circumvent this problem by appropriately designing the external force vector in (8) so that the first and second terms on the right-hand side of (8) cancel out each other. This ensures CoP tracking without the need for extensive controller action, as depicted in Fig. 1. The idea is to abandon the constant CAM assumption and plan the eCMP trajectory, and thus the external force vector (3), based on the actual multibody CAM.

As reported in [28], the eCMP is given by the CoP position and an offset depending on the horizontal torque about the CoM[2]

$$\boldsymbol{e} = \boldsymbol{p} + \frac{1}{m(g+\ddot{z})} \begin{pmatrix} \tau_{c,y} \\ -\tau_{c,x} \\ 0 \end{pmatrix}. \tag{10}$$

To provide explicit solutions, the eCMP trajectory needs to be expressed as a polynomial function, as introduced in (4). Inspired by the approach in [15], we combine (4) and (10) and represent the desired CoP and the torque about the CoM by individual polynomial functions. This yields

$$\boldsymbol{e}(t) = \sum_{j=0}^{n_p} \boldsymbol{a}_{p,j} t^j + \frac{1}{mg} \sum_{j=0}^{n_\tau} \boldsymbol{a}_{\tau,j} t^j, \tag{11}$$

where n_p, n_τ, $\boldsymbol{a}_{p,j} \in \mathbb{R}^3$ and $\boldsymbol{a}_{\tau,j} \in \mathbb{R}^3$ are the degrees and the coefficients of the CoP and CoM torque polynomial, respectively. The derivation of the coefficients in (11) is discussed in the following section. Note that hereafter, flat-ground walking with $\ddot{z} = 0$ is assumed. Alternatively, for walking with a non-constant CoM height, the term $\frac{1}{m(g+\ddot{z})}$ can be expressed by an additional polynomial function.

[2]In this work, the eCMP coincides with the CMP since the DCM time constant is computed based on the average CoM height above the ground.

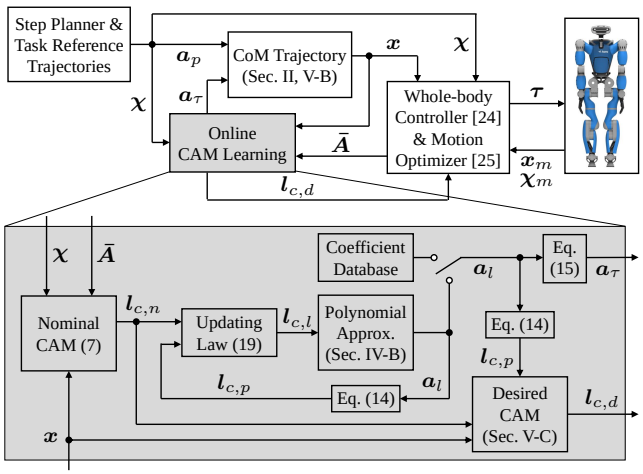


Fig. 2. Overview of the system architecture. For brevity, time derivatives of system states are omitted.

IV. Polynomial VRP Reference

The VRP reference is generated based on a sequence of footsteps given as foot centers and a learned CAM trajectory expressed as polynomials. The corresponding polynomial coefficients $\boldsymbol{a}_p$ and $\boldsymbol{a}_\tau$ are grouped into $\boldsymbol{P}_v$ and used in Section II-B to compute the CoM trajectory.

A. CoP Coefficients

The CoP reference is obtained from a spatial linear interpolation between a sequence of desired CoP positions, which are designed to coincide with the foot centers of the planned footsteps

$$\boldsymbol{p}_d(t) = (1 - f(t))\boldsymbol{p}_{d,0} + f(t)\boldsymbol{p}_{d,T}. \tag{12}$$

The start point of the current transition phase is given by $\boldsymbol{p}_{d,0}$, while the end point $\boldsymbol{p}_{d,T}$ is chosen to be the start point of the following transition phase. The temporal interpolation function $f(t)$ is a polynomial of degree n_p and designed to ensure continuity of the CoP reference at phase transitions. For example, a temporal linear interpolation ($n_p = 1$) in vectorized form yields

$$\boldsymbol{p}_d(t) = \underbrace{\begin{bmatrix} \boldsymbol{p}_{d,0}, & \frac{1}{T_\varphi}(\boldsymbol{p}_{d,T} - \boldsymbol{p}_{d,0}) \end{bmatrix}}_{\boldsymbol{a}_p} \begin{pmatrix} 1 \\ t \end{pmatrix}. \tag{13}$$

An additional heel-to-toe motion can be implemented by adding further CoP waypoints.

B. CAM Coefficients

The goal is to incorporate the rate of change of CAM, induced by the multibody dynamics of the robot, in the closed-form solutions of the DCM framework. Therefore, we approximate the CAM through polynomial functions, which can be used straightforwardly in (11) to generate a VRP reference. In [15], the multibody CAM is approximated with a simplified three-mass model, which gives only a rough estimate of the actual dynamics. In contrast to that, we use

an online learned CAM trajectory to obtain a realistic representation of the CAM induced by the multibody dynamics of the robot. For each transition phase, a polynomial function $\boldsymbol{l}_{c,p}(t)$ of degree $n_l = n_\tau + 1$ is fit online to a learned CAM trajectory $\boldsymbol{l}_{c,l}(t)$ that is obtained through online recording of the CAM of previous walking sequences (see Section V-A).

The polynomial function approximating the CAM has the form

$$\boldsymbol{l}_{c,p}(t) = \sum_{j=0}^{n_l} \boldsymbol{a}_{l,j} t^j \tag{14}$$

with $\boldsymbol{a}_{l,j} \in \mathbb{R}^3$ and the local time $t \in [0,\ T_\varphi]$. The corresponding rate of change, i.e., the torque about the CoM, is given by

$$\dot{\boldsymbol{l}}_{c,p}(t) = \sum_{j=0}^{n_\tau} (j+1)\boldsymbol{a}_{l,j+1} t^j = \sum_{j=0}^{n_\tau} \boldsymbol{a}_{\tau,j} t^j. \tag{15}$$

To obtain the polynomial coefficients $\boldsymbol{a}_l$, the given $\boldsymbol{l}_{c,l}$ is discretized for each transition phase into single data points with a discretized local time $t_h = h\Delta t$ and $h \in \{1, ..., k\}$. Here, Δt is the time interval between individual data points and needs to be a multiple of the sampling time Δt_s of the system and $k = \lfloor T_\varphi / \Delta t \rfloor$ is the number of data points. For each data point, we obtain an equation based on (14). Finally, we can write a linear system of k equations for the entire transition phase as follows:

$$\underbrace{\begin{bmatrix} t_k^{n_l} & t_k^{n_l-1} & \dots & 1 \\ t_{k-1}^{n_l} & t_{k-1}^{n_l-1} & \dots & 1 \\ \vdots & & & \\ t_1^{n_l} & t_1^{n_l-1} & \dots & 1 \end{bmatrix}}_{\bar{\boldsymbol{V}}} \underbrace{\begin{bmatrix} \boldsymbol{a}_{l,n_l}^T \\ \boldsymbol{a}_{l,n_l-1}^T \\ \vdots \\ \boldsymbol{a}_{l,0}^T \end{bmatrix}}_{\boldsymbol{a}_l^T} = \underbrace{\begin{bmatrix} \boldsymbol{l}_{c,l}^T(t_k) \\ \boldsymbol{l}_{c,l}^T(t_{k-1}) \\ \vdots \\ \boldsymbol{l}_{c,l}^T(t_1) \end{bmatrix}}_{\bar{\boldsymbol{l}}_{c,l}^T}. \tag{16}$$

Here, $\bar{\boldsymbol{V}} \in \mathbb{R}^{k \times (n_l+1)}$ is a so-called Vandermonde matrix. The following boundary conditions are chosen to ensure continuity between transition phases: $\boldsymbol{l}_{c,p}(0) = \boldsymbol{l}_{c,l}(0)$, $\dot{\boldsymbol{l}}_{c,p}(0) = \dot{\boldsymbol{l}}_{c,l}(0)$, $\boldsymbol{l}_{c,p}(T_\varphi) = \boldsymbol{l}_{c,l}(T_\varphi)$, $\dot{\boldsymbol{l}}_{c,p}(T_\varphi) = \dot{\boldsymbol{l}}_{c,l}(T_\varphi)$. We insert the boundary conditions into (16) and obtain a relationship for $\boldsymbol{a}_{l,4:n_l}$, the $n_l - 3$ last elements of the unknown polynomial coefficient matrix

$$\bar{\boldsymbol{V}}_1 \boldsymbol{a}_{l,4:n_l}^T + \bar{\boldsymbol{V}}_2 = \bar{\boldsymbol{l}}_{c,l}^T. \tag{17}$$

The matrices $\bar{\boldsymbol{V}}_1 \in \mathbb{R}^{k \times (n_l-3)}$ and $\bar{\boldsymbol{V}}_2 \in \mathbb{R}^{k \times 3}$ are only a function of the local time of each data point, the total duration of the transition phase, and the boundary conditions. We are interested in a good approximation of $\boldsymbol{l}_{c,l}$ over the entire transition phase and therefore chose to have considerably more data points than unknown polynomial coefficients in $\boldsymbol{a}_{l,4:n_l}$, i.e., $k > n_l - 3$. Formulating the least-square solution of the overdetermined set of equations in (17) yields:

$$\boldsymbol{a}_{l,4:n_l}^T = (\bar{\boldsymbol{V}}_1^T \bar{\boldsymbol{V}}_1)^{-1} \bar{\boldsymbol{V}}_1^T (\bar{\boldsymbol{l}}_{c,l}^T - \bar{\boldsymbol{V}}_2). \tag{18}$$

To reduce the computational cost, the matrix $(\bar{\boldsymbol{V}}_1^T \bar{\boldsymbol{V}}_1)^{-1} \bar{\boldsymbol{V}}_1^T$ can be precomputed offline for given values of Δt and T_φ. The remaining unknown polynomial coefficients $\boldsymbol{a}_{l,0:3} = [\boldsymbol{a}_{l,3}^T, ..., \boldsymbol{a}_{l,0}^T]^T$ are computed based on $\boldsymbol{a}_{l,4:n_l}$ and the boundary conditions.

Finally, $\boldsymbol{P}_v$ is obtained by grouping the coefficient matrices $\boldsymbol{a}_p$ and $\boldsymbol{a}_l$ and adding the constant vertical offset Δz. Notice from (10) that only the CoM torque within the plane parallel to the support surface (xy-direction) is needed, and thus only the corresponding coefficients are computed. The coefficients for the CoM torque about the z-axis are set to zero.

V. Online Learning of CAM and Trajectory Updating Scheme

This section presents a method for online learning of the nominal CAM $\boldsymbol{l}_{c,n}$, which is used to obtain the polynomial coefficient matrix $\boldsymbol{a}_l$ in Section IV-B. Moreover, a desired CAM trajectory $\boldsymbol{l}_{c,d}$ is generated, which is explicitly tracked by the whole-body controller.

A. Online Learning of the Nominal CAM Trajectory

We aim to learn a CAM trajectory $\boldsymbol{l}_{c,l}$ based on online recorded CAM values. Since walking is a repetitive process, our proposed method iteratively records the model-based computed nominal CAM over successive iterations. One iteration consists of four transition phases, i.e., two single and two double support phases. The nominal CAM $\boldsymbol{l}_{c,n,i}$ of iteration i is obtained from (7) using the CoM velocity, the CMM, and the reference task-space velocity vector obtained from the planner of the corresponding iteration. The following iterative scheme is introduced to update the learned CAM:

$$\boldsymbol{l}_{c,l,i+1}(t) = (1 - k_l)\, \boldsymbol{l}_{c,p,i}(t) + k_l\, \boldsymbol{l}_{c,n,i}(t). \tag{19}$$

Here, $0 < k_l \leq 1$ is the learning factor and $\boldsymbol{l}_{c,p,i}$ is the polynomial function approximating the learned CAM $\boldsymbol{l}_{c,l,i}$ during the i^{th} iteration.

At the end of each transition phase within one iteration, $\boldsymbol{l}_{c,p,i+1}$ is obtained by applying the polynomial fitting algorithm of Section IV-B to $\boldsymbol{l}_{c,l,i+1}$ for the respective transition phase. The updating procedure is inspired by a Run by Run controller [29], which updates input parameters between iterations based on measurements made while the iteration is running.

Note that learning the nominal CAM only during the first iteration is not sufficient since the updated CoM trajectory leads to changes in the robot configuration, affecting the CAM of the following iteration. An exemplary convergence behavior between the nominal and polynomial approximated CAM is shown in Fig. 6a. Larger values of k_l in (19) lead to faster convergence, while smaller values lead to more robust convergence. In particular, if the robot deviates strongly from its reference configuration in the first iterations due to model uncertainties or tracking errors, a smaller learning factor is beneficial to avoid overcompensation.

The learned polynomial coefficients $\boldsymbol{a}_l$ are always associated with a certain set of walking parameters. As presented in [18], the obtained knowledge can be used to build up a coefficient database of typical walking parameters, which can also be pre-trained in simulation.

B. Online Updating of the DCM Trajectory

In the initial iteration (iteration ⓪), the polynomial coefficient matrix $\boldsymbol{a}_l$ is initialized with zero, and the entire DCM reference trajectory is computed. This is equivalent to generating the DCM trajectory with a constant CAM assumption. Note that if values for $\boldsymbol{a}_l$ from previous runs are available in the coefficient database, these can be used instead to speed up convergence. The DCM backward computation scheme is applied at the end of each transition phase φ, starting from the end of transition phase $\varphi+4$ and using the updated polynomial coefficients of iteration $i+1$. As terminal DCM constraint for the backward computation, the terminal DCM at the end of phase $\varphi+4$ of the initial iteration ($\boldsymbol{\xi}_{T,\varphi+4,0}$) is used. The CAM knowledge obtained during the current iteration is directly applied in the following one.

To ensure continuity of the updated DCM reference at the start of phase $\varphi+1$, the final DCM of phase $\varphi+1$ is linearly interpolated between its values during the current and next iteration

$$\boldsymbol{\xi}_{T,\varphi+1}(t) = \Big(1-\frac{t}{T_{\varphi+1}}\Big)\boldsymbol{\xi}_{T,\varphi+1,i} + \frac{t}{T_{\varphi+1}}\boldsymbol{\xi}_{T,\varphi+1,i+1}. \quad (20)$$

C. Generating a Desired CAM Trajectory

In this section, we generate a desired CAM trajectory which is commanded to the whole-body controller to ensure contact torque tracking. For the xy-direction, CoP tracking can be achieved by choosing the desired CAM equal to the polynomial approximated CAM used within CoM trajectory generation $\boldsymbol{l}_{c,d,xy} = \boldsymbol{l}_{c,p,xy}$. However, during transition phases where $\boldsymbol{a}_l$ and thus $\boldsymbol{l}_{c,p}$ is zero, e.g., in the initial iteration, starting or stopping phase, the desired CAM is chosen as the nominal CAM $\boldsymbol{l}_{c,d,xy} = \boldsymbol{l}_{c,n,xy}$. Otherwise, the whole-body controller needs to regulate the entire CAM induced through the swing leg motion to zero, resulting in extensive upper-body motion.

Recalling (8), we obtain for the contact torque about the z-axis

$$\tau_{grf,z} = m(x-p_{d,x})\ddot{y} - m(y-p_{d,y})\ddot{x} + \dot{l}_{c,z} \quad (21)$$

with $\boldsymbol{x} = (x,\, y,\, z)^T$ and $\ddot{\boldsymbol{x}} = (\ddot{x},\, \ddot{y},\, \ddot{z})^T$ being the CoM position and acceleration, respectively. A pure regulation of $\dot{l}_{c,z}$, as put forward in [21]–[23], is not sufficient here since the terms resulting from the cross product are not vanishing because the external force vector is not planned to intersect the desired CoP and CoM position. To compensate the cross product terms and ensure $\tau_{grf,z} = 0$, the desired rate of change of CAM about the z-axis is defined as

$$\dot{l}_{c,d,z} = -m(x-p_{d,x})\ddot{y} + m(y-p_{d,y})\ddot{x} + \dot{l}^{cor}_{c,z}. \quad (22)$$

The corresponding CAM can be obtained by numerical integration. To ensure a symmetrical motion of the robot, e.g., after the transition between different walking speeds, a correction term $\dot{l}^{cor}_{c,z}$ is added. This term is introduced to bring the mean value of $l_{c,d,z}$ over an iteration to zero. It is computed based on the mean value of $l_{c,d,z}$ of the previous iteration.

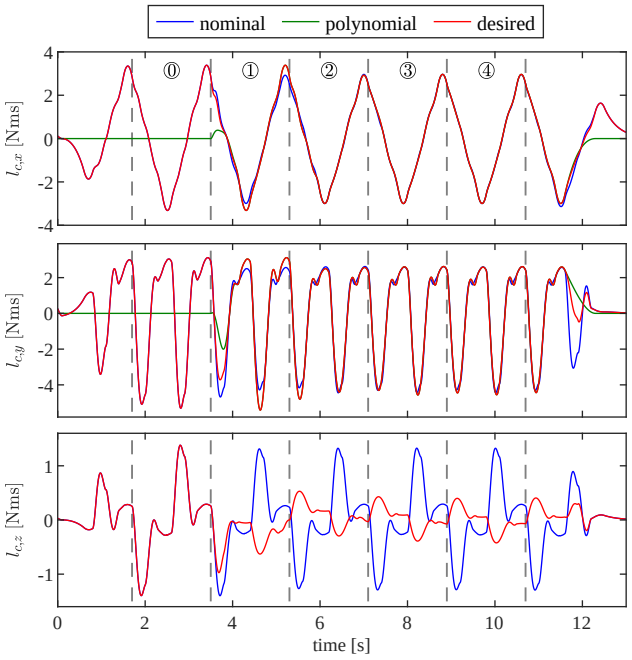


Fig. 3. Results of the experiment #1: Nominal ($\boldsymbol{l}_{c,n}$), polynomial approximated ($\boldsymbol{l}_{c,p}$) and desired ($\boldsymbol{l}_{c,d}$) CAM trajectories. The vertical dashed lines indicate iteration ⓪ to ④.

VI. Experimental Evaluation

The proposed algorithm is validated through experiments with the torque-controlled humanoid robot TORO [30]. The robot has 27 degrees of freedom, a height of 1.74 m, a total weight of 79.2 kg, and an inertial measurement unit (IMU) in the torso. Footage of the presented experiments can be found in the supplementary video.

We present two scenarios to evaluate the performance of our algorithm. In the first experiment, the robot walks 12 steps with a step length of 15 cm, which corresponds to a stride length of 30 cm. The single support time is $T_{SS} = 0.7$ s and the double support time is $T_{DS} = 0.2$ s. These walking parameters correspond to five complete iterations with an iteration time of $T_{iter} = 1.8$ s. It is assumed that no learned polynomial coefficients of previous runs are available, i.e., $\boldsymbol{a}_l = \boldsymbol{0}$.

Fig. 3 shows the nominal CAM obtained from (7), the polynomial approximated CAM used for the CoM trajectory generation, and the desired CAM, which is commanded to the whole-body controller. During iteration ⓪, $\boldsymbol{l}_{c,p,xy}$ is zero and convergences to $\boldsymbol{l}_{c,n,xy}$ in the following iterations. A polynomial of 9^{th}-order ($n_l = 9$) is used for the approximation of $\boldsymbol{l}_{c,n,xy}$. A discretization time interval of $\Delta t = 50$ ms is chosen, while the sampling frequency of the system is $\Delta t_s = 1$ ms. The learning factor is set to $k_l = 1$, and the corresponding convergence behavior is plotted in Fig. 6a. The nominal CAM varies from iteration to iteration since the updated CoM trajectory influences the robot configuration and thus the resulting CAM.

Starting from iteration ①, the influence of the online learned CAM on the VRP trajectory can be observed

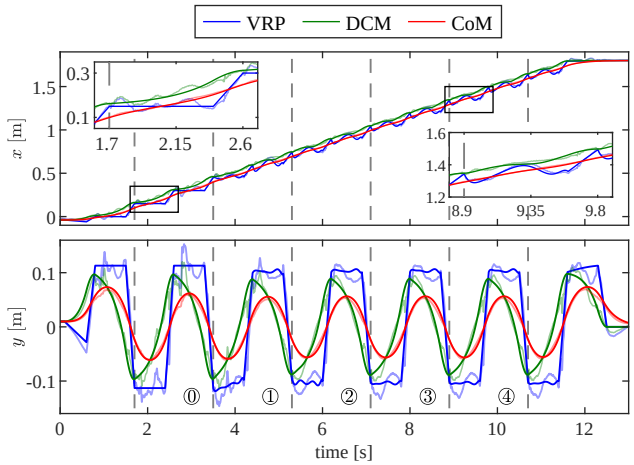


Fig. 4. Results of the experiment #1: Online updated VRP ($\boldsymbol{v}$), DCM ($\boldsymbol{\xi}$) and CoM ($\boldsymbol{x}$) trajectories. The darker colors represent the desired values, while the lighter ones show the measured values. The vertical dashed lines indicate iteration ⓪ to ④.

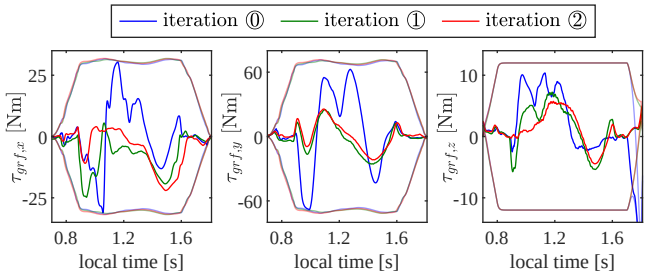
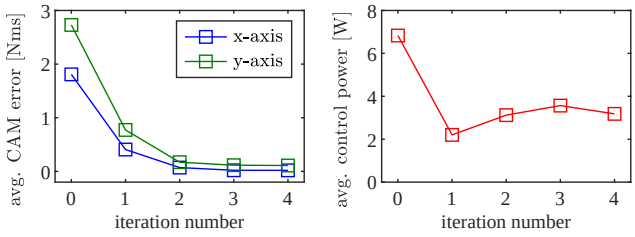


Fig. 5. Results of the experiment #1: Contact torques at the desired CoP position of the left foot in single support for several iterations given in local iteration time. Limits are indicated by the respective lighter colored lines.

in Fig. 4. The corresponding DCM and CoM trajectories are computed following the updating scheme from Section V-B. The darker variants of the respective color show the desired values, while the lighter ones represent the measured values.

The resulting contact torques at the desired CoP position are depicted in Fig. 5. Iteration ⓪ can be seen as baseline behavior for walking without our presented approach since no CAM knowledge is available during this iteration. Through the updated VRP reference and the appropriate design of the desired CAM trajectory, the contact torques are substantially reduced from iteration ① onwards. Without real world imperfections, the contact torques would be reduced to zero; however, model uncertainties and tracking errors degrade the result. The torque limits in xy-direction are computed based on the CoP constraint. Since the whole-body controller uses a wrench formulation, the torque limits about the z-axis are approximated by upper and lower values, for further details please refer to [24].

The reduced contact torques about all three axes lead to a more robust walking behavior, which can be observed in the supplementary video. During iteration ⓪, the feet of the robot tilt and twist slightly since the CoP reaches the edge of the support area in x- and y-direction and the torque limits about the z-axis are activated. In the following iterations, no violation of the contact constraints occurs, confirming



(a) (b)

Fig. 6. Results of the experiment #1: a) Average approximation error per iteration between nominal and polynomial CAM, i.e., $\boldsymbol{err}_i = \frac{1}{T_{iter}} \int_0^{T_{iter}} |\boldsymbol{l}_{c,n,i}(t) - \boldsymbol{l}_{c,p,i}(t)| \, \mathrm{d}t$; b) average control input power per iteration, i.e., $pow_i = \frac{1}{T_{iter}} \int_0^{T_{iter}} \dot{\boldsymbol{q}}_i(t)^T \boldsymbol{\tau}_i(t) \, \mathrm{d}t$. The control input power is the product of joint velocities and torques.

the increased contact robustness. Starting from iteration ①, upper-body motions are induced by explicitly tracking the desired CAM trajectory.

As the contact torques reach their limits during iteration ⓪, the whole-body controller cannot fully generate the desired CoM wrench, which degrades the tracking of the CoM dynamics. As shown in Fig. 4, the tracking of the CoM dynamics improves due to the reduced contact torques from iteration ① onwards.

In addition to a more robust walking motion, the inclusion of CAM during the CoM trajectory generation also leads to more efficient locomotion. As presented in Fig. 6b, the average control input power per iteration is considerably reduced after the initial iteration.

In the second experiment, shown in the supplementary video, the step length is increased to 22 cm while the other walking parameters remain unchanged. This improved performance can only be achieved by using polynomial coefficients pre-trained in simulation. As also reported in [19], TORO does not walk reliably with a step length of 15 cm and the timing parameters of experiment #1 without additional features like toe-off motions. Our approach enables flat foot walking with a step length of 22 cm, which is an increase of more than 50%, while using the identical walking parameters and whole-body controller.

VII. Conclusion

This work presented a method to reduce the contact torques at the desired CoP position through online learning of the CAM induced by the multibody dynamics. A polynomial function was fit to the learned data, and the corresponding coefficients were used to generate CoM trajectories in closed-form based on the concept of DCM. In addition, we introduced an online updating scheme to ensure the continuity of all references. Finally, a desired CAM trajectory was generated and tracked by the whole-body controller of the robot. The proposed method was validated in experiments. We demonstrated a more robust and efficient walking motion by enabling increased step length and faster walking speed.

As part of future research, the presented method will be extended to model-free learning of the CAM based on measurement data.

REFERENCES

[1] 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 1, 2001, pp. 239–246.

[2] M. Vukobratović and J. Stepanenko, "On the stability of anthropomorphic systems," *Mathematical Biosciences*, vol. 15, no. 1, pp. 1–37, 1972.

[3] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa, "An analytical method on real-time gait planning for a humanoid robot," in *Proc. 4th IEEE-RAS Int. Conf. Humanoid Robots*, vol. 2, 2004, pp. 640–655.

[4] R. Tedrake, S. Kuindersma, R. Deits, and K. Miura, "A closed-form solution for real-time ZMP gait generation and feedback stabilization," in *Proc. 15th IEEE-RAS Int. Conf. Humanoid Robots*, 2015, pp. 936–940.

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

[6] T. Takenaka, T. Matsumoto, and T. Yoshiike, "Real time motion generation and control for biped robot -1st report: Walking gait pattern generation-," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 1084–1091.

[7] J. Pratt, T. Koolen, T. de Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson, and P. Neuhaus, "Capturability-based analysis and control of legged locomotion, part 2: Application to M2V2, a lower-body humanoid," *Int. J. Robot. Res.*, vol. 31, no. 10, pp. 1117–1133, 2012.

[8] J. Englsberger, 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.

[9] G. Mesesan, J. Englsberger, 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.

[10] J. Englsberger, G. Mesesan, C. Ott, and A. Albu-Schäffer, "DCM-based gait generation for walking on moving support surfaces," in *Proc. 18th IEEE-RAS Int. Conf. Humanoid Robots*, 2018, pp. 1–8.

[11] A. Herzog, N. Rotella, S. Schaal, and L. Righetti, "Trajectory generation for multi-contact momentum control," in *Proc. 15th IEEE-RAS Int. Conf. Humanoid Robots*, 2015, pp. 874–880.

[12] 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.

[13] S.-H. Lee and A. Goswami, "Reaction mass pendulum (RMP): An explicit model for centroidal angular momentum of humanoid robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 4667–4672.

[14] K. Guan, K. Yamamoto, and Y. Nakamura, "Virtual-mass-ellipsoid inverted pendulum model and its applications to 3d bipedal locomotion on uneven terrains," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 1401–1406.

[15] T. Seyde, A. Shrivastava, J. Englsberger, S. Bertrand, J. Pratt, and R. J. Griffin, "Inclusion of angular momentum during planning for capture point based walking," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 1791–1798.

[16] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, 2003, pp. 1620–1626.

[17] K. Nishiwaki and S. Kagami, "Online walking control system for humanoids with short cycle pattern generation," *Int. J. Robot. Res.*, vol. 28, no. 6, pp. 729–742, 2009.

[18] K. Hu, C. Ott, and D. Lee, "Learning and generalization of compensative zero-moment point trajectory for biped walking," *IEEE Trans. on Robotics*, vol. 32, no. 3, pp. 717–725, 2016.

[19] S. Wang, G. Mesesan, J. Englsberger, D. Lee, and C. Ott, "Online virtual repellent point adaptation for biped walking using iterative learning control," in *Proc. 20th IEEE-RAS Int. Conf. Humanoid Robots*, 2021, pp. 112–119.

[20] P. Sardain and G. Bessonnet, "Forces acting on a biped robot. center of pressure-zero moment point," *IEEE Trans. on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 34, no. 5, pp. 630–637, 2004.

[21] A. Miyata, S. Miyahara, and D. N. Nenchev, "Walking with arm swinging and pelvis rotation generated with the relative angular acceleration," *IEEE Robot. Autom. Lett.*, vol. 5, no. 1, pp. 151–158, 2020.

[22] M. Schwienbacher, T. Buschmann, S. Lohmeier, V. Favot, and H. Ulbrich, "Self-collision avoidance and angular momentum compensation for a biped humanoid robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 581–586.

[23] B. Park, M. Kim, E. Sung, J. Kim, and J. Park, "Whole-body walking pattern using pelvis-rotation for long stride and arm swing for yaw angular momentum compensation," in *Proc. 20th IEEE-RAS Int. Conf. Humanoid Robots*, 2021, pp. 47–52.

[24] G. Mesesan, J. Englsberger, G. Garofalo, C. Ott, and A. Albu-Schäffer, "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.

[25] R. Schuller, G. Mesesan, J. Englsberger, J. Lee, and C. Ott, "Online centroidal angular momentum reference generation and motion optimization for humanoid push recovery," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5689–5696, 2021.

[26] M. B. Popovic, A. Goswami, and H. Herr, "Ground reference points in legged locomotion: Definitions, biological trajectories and control implications," *Int. J. Robot. Res.*, vol. 24, no. 12, pp. 1013–1032, 2005.

[27] 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.

[28] M. A. Hopkins, D. W. Hong, and A. Leonessa, "Humanoid locomotion on uneven terrain using the time-varying divergent component of motion," in *Proc. 14th IEEE-RAS Int. Conf. Humanoid Robots*, 2014, pp. 266–272.

[29] E. Sachs, R.-S. Guo, S. Ha, and A. Hu, "On-line process optimization and control using the sequential design of experiments," in *Digest of Technical Papers. Symposium on VLSI Technology*, 1990.

[30] J. Englsberger *et al.*, "Overview of the torque-controlled humanoid robot TORO," in *Proc. 14th IEEE-RAS Int. Conf. Humanoid Robots*, 2014, pp. 916–923.