Quintic Spline Collocation for Real-Time Biped Walking-Pattern Generation with variable Torso Height

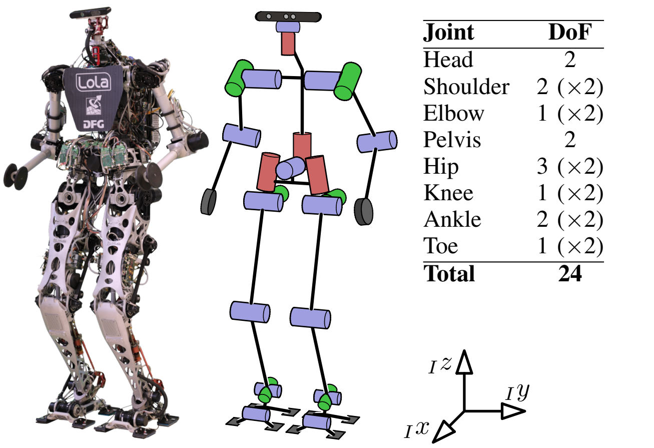
Philipp Seiwald, Felix Sygulla, Nora-Sophie Staufenberg and Daniel Rixen[1]

Abstract— **This paper presents our newest findings in planning a dynamically and kinematically feasible center of mass motion for bipedal walking robots. We use a simplified robot model to incorporate multi-body dynamics and kinematic limits, while still being able to meet hard real-time requirements. The vertical center of mass motion is obtained through interpolation of a quintic spline whose control points are projected onto the kinematically feasible region. Subsequently, the horizontal motion is computed from multi-body dynamics which we approximate by solving an overdetermined boundary value problem via spline collocation based on quintic polynomials. The proposed algorithm is an improvement of our previous method, which used a parametric torso height optimization for vertical and cubic spline collocation for horizontal components. The novel center of mass motion improves stability, especially for stepping up and down platforms. Moreover, the new method leads to a less complex overall algorithm since it removes the necessity of manually tuned parameters and strongly simplifies the incorporation of boundary values. Lastly, the new approach is more efficient, which leads to a significantly reduced total runtime. The proposed method is validated through successfully conducted simulations and experiments on our humanoid robot platform, LOLA.**

I. INTRODUCTION

In the recent past, the capabilities of bipedal robots, especially humanoids, have experienced tremendous progress. Although the motion of some prototypes may still look wobbly when compared to humans as a point of reference, as seen in the DRC [1], the robustness and versatility of locomotion has steadily improved. How close today's systems are to real-life applications, assuming high-end hardware and appropriate manpower for fine-tuning, is demonstrated, among others, by *Boston Dynamics* and *Honda* with their recent versions of ATLAS [2] and ASIMO [3].

A challenge which comes hand in hand with growing skills is the increasing complexity of the software framework and its underlying algorithms. In our opinion, this is a severe issue harming usability and constraining further research. Thus, from time to time, one should condense the experience gained through using complex algorithms to identify the essential behavior, and try to replace them with simpler ones. In accordance with this idea, the proposed method aims at improving, simplifying, and generalizing our previous methods. In particular, we present in Section IV an algorithm for generating a dynamically and kinematically feasible *Center of Mass (CoM)* motion for bipedal robots, given a set of discrete footsteps. In contrast to the previously used spline collocation method in [4], which relies on cubic polynomials, the presented approach is based on quintic polynomials, which may seem to increase complexity at first sight. However, this modification allows a straightforward integration of boundary values which in turn lowers overall complexity. Additionally, runtime is reduced, since fewer collocation sites are necessary to obtain the same approximation quality, cf. [5]. Furthermore, we avoid manually tuned parameters of the torso height optimization in [6], by following a simpler, pure geometric approach to incorporate kinematic limits.



Joint	DoF
Head	2
Shoulder	2 (×2)
Elbow	1 (×2)
Pelvis	2
Hip	3 (×2)
Knee	1 (×2)
Ankle	2 (×2)
Toe	1 (×2)
Total	**24**

Fig. 1. The humanoid robot LOLA and its "full" kinematic structure. The 24 DoF system is 180 cm tall and weighs approximately 60 kg.

In Section II, we briefly discuss other approaches for planning feasible CoM (and whole body) motion. Certainly, the overall walking performance heavily depends on the hardware and surrounding software framework which varies for each system. For this reason, in Section III, we give a brief introduction to our test platform LOLA, which was used to obtain the results discussed in Section V.

II. BACKGROUND AND RELATED WORK

Just like the majority of walking-pattern generators for bipeds, the proposed method is based on the *Zero-Moment Point (ZMP)* concept [7] to create a dynamically balanced gait. Within this contribution, we avoid a tilting motion, such that the ZMP is equivalent to the *Center of Pressure (CoP)* and the *Fictious Zero-Moment Point (FZMP)* [8].

Various algorithms implement a tracking controller to realize a predefined ZMP motion. For example the gradient descent method is used in [9] to minimize ZMP error by modifying the horizontal torso motion. The necessity to evaluate the full multi-body dynamics within each iteration leads to execution times which are impractical for online

[1] Technical University of Munich, Chair of Applied Mechanics, 85748 Garching, Germany. E-Mail: philipp.seiwald@tum.de

generation. Real-time planning was achieved in [10], where the motion of all body parts, except the feet, is modified in the horizontal plane and the *Boundary Value Problem (BVP)* is solved using a finite difference method. In contrast to our proposed method, the algorithm in [10] requires a feasible initial solution which is then optimized in an iterative scheme. In order to maintain real-time performance while preserving the essential dynamics of the system, various simplified models have been introduced. In [11], it was shown that biped robots may be approximated by a simple *Linear Inverted Pendulum Mode (LIPM)* [12]. The model has been extended to the three-dimensional case in [13] which was then coupled in [14] with the ZMP concept through the formulation as an inverse dynamics problem. To consider swing-foot dynamics, the LIPM has been extended by additional point-masses for the feet, cf. [15]. TAKENAKA et al. additionally separated the dynamics of the LIPM in the convergent and the *Divergent Component of Motion (DCM)*. The ZMP is then modified to generate a cyclic gait for the DCM. This allowed impressive walking speeds of up to 4 km/h. An extension of the DCM concept to the 3D case is given in [16] where a prescribed, so-called *Virtual Repellent Point* – as coupling point of a linear spring representing the ground reaction and gravity force – defines the dynamics of the DCM, which in turn specifies the planned CoM motion. Although the assumption of instantaneous leg switch was dropped in [17], various other simplifications, e. g. point-like feet, were necessary to obtain an analytical solution.

In [4], a more generic three-mass model is used, where the torso is approximated by a point mass with arbitrary, but predefined, vertical motion. The BVP describing the horizontal torso motion is solved using a spline collocation method which is also the basis of our proposed algorithm. The main advantage of this method is that there is no restriction as to the shape of the ZMP or vertical torso motion, such that it can be used to incorporate kinematic constraints as shown in [6] and adapted in this paper. For the sake of brevity, we do not list methods for spline collocation here, but instead refer to [5], which collects literature related to our proposed algorithm. Another extension to the LIPM is made in [18], where an additional flywheel together with actuation constraints are used to compute a region of so-called *Capture Points (CPs)*, i. e. predicted footholds which bring the robot to a complete stop. Although the main motivation in [18] is push recovery, extensions to use CPs for walking-pattern generation have been discussed, e. g. in [19]. Note that, while the CP and DCM are strongly related to each other in 2D, this does not hold true in 3D [17].

Most of the mentioned walking-pattern generators do not consider kinematic limits. However, especially for complex motions, e. g. stepping over obstacles or climbing stairs, certain constraints have to be respected. Our approach tackles this problem by varying the torso height accordingly. This strategy was also investigated (among other works) in [20] and [21]. In [20], the vertical waist trajectory is built by third-order polynomials connected by a so-called *key configuration*, which is obtained from the obstacle's geometry. Subsequently, the preview control from [14] is used to compute the horizontal components (although the underlying LIPM implies a constant CoM height). In contrast to [20], our method is based on a three-mass model, which incorporates the dynamic effects of a varying torso height by construction. Similar to [21], we use simplified leg kinematics to compute limits for vertical torso motion, however, in contrast to [21], our method also considers the lower bound.

An approach generally unrelated to the ZMP is proposed in [22] where whole-body motion is planned by formulating a comprehensive optimization problem. In the context of multi-contact locomotion, feasibility and balance is achieved by considering the contact wrench cone together with various constraints to avoid kinematic limits and collisions. Although complex tasks have been demonstrated, the large number of optimization variables and constraints leads to high computational cost, which makes real-time execution difficult.

III. SYSTEM OVERVIEW

Although real-time walking-pattern generation has been achieved almost two decades ago, only a small amount of modern full-size humanoids are capable of robust and fast locomotion. In our opinion, the interplay of potent hardware and a robust software framework is crucial.

A. Hardware

The kinematic structure of LOLA features 24 electrically actuated *Degrees of Freedom (DoFs)*, cf. Figure 1. All joints are driven by brushless DC motors, where most of them incorporate high-ratio harmonic gearing. The knee and ankle joints feature planetary roller screws and slider crank mechanisms, respectively, to improve load rating while minimizing mass and inertia. All joints are back-drivable and show only minimal backlash. A detailed overview of the mechanical design of the robot is given in [23]. Each foot integrates a custom six-axis force-torque sensor and four binary contact switches to detect early- and late-contact events. The torso contains a high precision *inertial measurement unit (IMU)* and two mainboards, each with an Intel Core i7-4770S@3.1 GHz (4x) processor and 8GB RAM. On the first board, the real-time operating system QNX NEUTRINO 6.6 runs all planning and stabilization algorithms together with the hardware abstraction layer. The second board is dedicated to the vision system, which is not directly related to the algorithms investigated in this contribution, thus a further description is omitted. The communication between the distributed sensors, commercial servo controllers and the control algorithms of the first onboard PC is effected via an ETHERCAT bus. A comprehensive description of the electromechanical components and the communication system of LOLA is given in [24].

B. Software

The software and control framework of LOLA is depicted in Figure 2. Through a graphical *User Interface*, high-level commands, such as walking direction, step length and duration, are sent to the *Walking-Pattern Generation (WPG)* module. Based on the perceived environment and simplified robot models, the WPG creates a motion plan consisting of *ideal task-space trajectories* which represent a dynamically balanced gait and include, among others,

- position $[x_F(t),\, y_F(t),\, z_F(t)]^T$, orientation[1] $\varphi_F(t)$ and toe-joint angles $\varphi_T(t)$ for both feet,
- torso orientation and CoM position.

A description of how these trajectories are created is given in Section IV. The ideal task-space trajectories are then passed to the *Stabilization and Inverse Kinematics (SIK)* module, which additionally takes into account direct sensor input to modify the planned motion and thus stabilizes the robot in the presence of disturbances. An overview of the implemented hybrid position/force control is given in [26]. For this paper, we use the version specified in [27] where support for partial contacts is disabled. The inverse kinematics in SIK is based on *Automatic Supervisory Control* [28] to avoid collisions and minimize various penalties like vertical angular momentum. Note that the arm motion is not part of the task-space definition, i. e., it is optimized in the null-space instead [29].

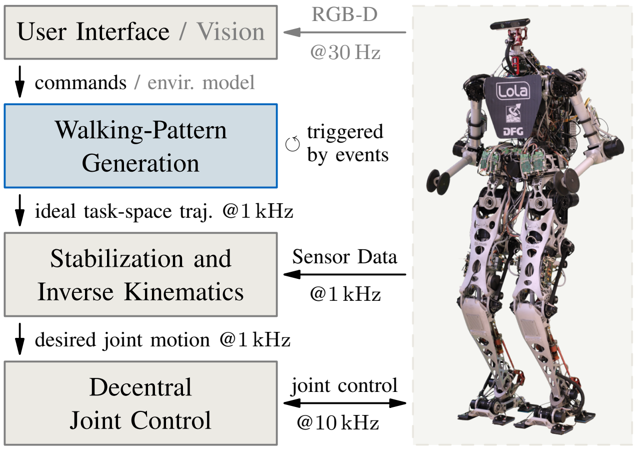


Fig. 2. Overview of the software/control framework of LOLA. The proposed algorithm is part of the *Walking-Pattern Generation (WPG)* module.

IV. PROPOSED ALGORITHM

In the following, we present a workflow to obtain the ideal task-space trajectories as output of the WPG module. The process can be subdivided into five stages, which are depicted in Figure 3. The *Reduced Model Torso (RMT)* point represents the torso center of our reduced model, which will be introduced later (cf. Figure 6). For the moment it should be considered as an intermediate result for obtaining the final CoM motion in Stage 5. Note that Stage 3 and 4 are run twice in order to resolve a circular dependency, which is explained later.

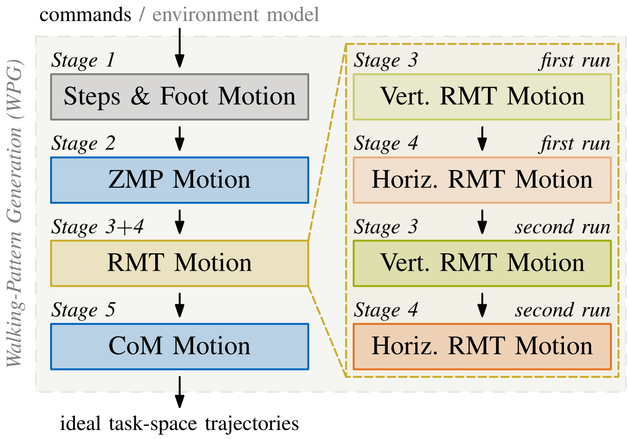


Fig. 3. Overview of the five proposed planning stages as components of the *Walking-Pattern Generation (WPG)* module. Core elements of the proposed algorithm are the generation of vertical and horizontal *Reduced Model Torso (RMT)* motion.

Stage 1: Foothold Sequence and Foot Motion

Given only high-level commands such as a target position, the robot tries to find a feasible path through its environment first. Subsequently, a sequence of foothold positions following this path is planned. Both can be done by path-planning algorithms with obstacle avoidance as proposed in [30]. Note that one may choose an arbitrary other algorithm as this does not constrain the following stages. For the sake of simplicity, we use predefined foothold sequences in the following.

As a second step, collision-free foot motions, consisting of trajectories for position, orientation and toe-joint angles, which connect the discrete footholds, are planned. Again, one may use an arbitrary algorithm for this, however, the position trajectories directly influence the CoM motion (cf. Stage 5), thus they must be $\mathcal{C}^2$-smooth in order to obtain a $\mathcal{C}^2$-smooth CoM trajectory. Here, we generate the foot motion as polynomials of maximum degree five, based on simple heuristics, cf. Figure 4. In particular, we modify the choice considered in [31] and use

$$
\begin{aligned}
\varphi_{F,\mathrm{max},i} &= \arcsin(0.1\,\mathrm{rad\,m}^{-1} \cdot L_{\mathrm{step},i}) \;, \\
z_{F,\mathrm{max},i} &= \max\{z_{F,i},\, z_{F,i+1}\} + \Delta z_{\mathrm{step}} + \Delta z_{F,\mathrm{comp},i} \;, \\
\varphi_{T,\mathrm{max},i} &= \Delta\varphi_{T,i,\leftrightarrow} + \Delta\varphi_{T,i,\uparrow}
\end{aligned}
$$

with the auxiliary variables $\Delta\varphi_{T,i,\leftrightarrow}$ and $\Delta\varphi_{T,i,\uparrow}$ given by

$$
\begin{aligned}
\Delta\varphi_{T,i,\leftrightarrow} &= \begin{cases} \frac{\pi}{18}\mathrm{rad} \cdot \frac{|L_{\mathrm{step},i}| - L_{\mathrm{foot}}}{0.8\,\mathrm{m} - L_{\mathrm{foot}}} & \text{for } |L_{\mathrm{step},i}| \geq 2\,L_{\mathrm{foot}} \;, \\ 0 & \text{else} \end{cases} \\
\Delta\varphi_{T,i,\uparrow} &= \max\{3\,\mathrm{rad\,m}^{-1} \cdot (z_{F,i+1} - z_{F,i})\,,\, 0\} \;.
\end{aligned}
$$

Here $\varphi_{F,\mathrm{max},i}$, $z_{F,\mathrm{max},i}$ and $\varphi_{T,\mathrm{max},i}$ denote the maximum rotation, vertical position and toe-joint angle of the current swing foot during the i-th physical step $t \in [t_i,\, t_{i+1}]$ with $T_{\mathrm{step}} = t_{i+1} - t_i$, respectively. Furthermore, $L_{\mathrm{step},i}$ is the i-th step length[2], L_{foot} the geometric length of the flat foot, Δz_{step} the desired step height (user-defined) and $\Delta z_{F,\mathrm{comp},i}$

[1] In the WPG module, orientations are parameterized using quaternions. For describing rotational motion we use SLERP interpolation [25] together with a time-dependent polynomial of maximum degree five, to customize the interpolation parameter of SLERP. Here, for the sake of brevity, we discuss only the two-dimensional projection, i. e., simple planar rotation.

[2] The i-th step length $L_{\mathrm{step},i}$ is defined as the traveled horizontal distance of the swing foot during single support of the i-th step.

a compensation for the vertical shift due to foot rotation. Note that we differentiate between foot rotation φ_F, which is defined as the toe segment inclination relative to the horizontal plane, and the toe-joint angle φ_T, which is the angle between the toe and heel segment, cf. Figure 6. In order to maximize the kinematic capabilities for large steps, φ_T depends on the step length, see $\Delta\varphi_{T,i,\leftrightarrow}$. With $\Delta\varphi_{T,i,\uparrow}$ we shift kinematic limits for stepping up. Note that we do, due to hardware restrictions, currently not use the toe-joint for stepping down. However, we plan to redesign the feet in the near future, thus allowing us to exploit the benefits of active toe-joints for stepping down. Finally, the choice of φ_F leads to the preferred heel-strike in case of an early contact event [31].

The horizontal components of the swing foot motion, i. e., $x_F(t)$ and $y_F(t)$, simply connect the discrete footholds. Note that foot rotation around the vertical axis (i. e., for walking on a curved path) is planned similarly to $x_F|y_F$, thus not further discussed. The resulting $\mathcal{C}^2$-smooth foot motion for one physical step T_{step}, consisting of the *Double Support (DS)* phase $(0.2 \cdot T_{\text{step}})$ and the *Single Support (SS)* phase $(0.8 \cdot T_{\text{step}})$, is shown in Figure 4.

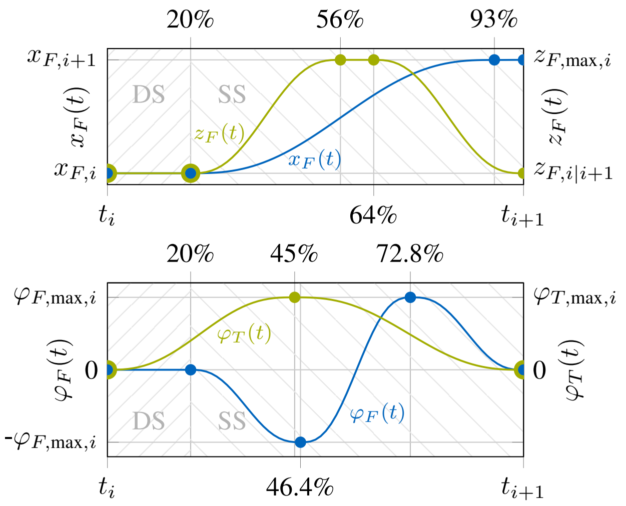


Fig. 4. *Normalized* swing foot position $x_F|z_F$, rotation φ_F and toe-joint angle φ_T for one physical step T_{step}. The stationary control points of the splines consisting of piecewise quintic polynomials are marked by dots.

Stage 2: Zero-Moment Point (ZMP) Motion

For a dynamically balanced gait, we constrain the ZMP to remain within the current *Support Polygon (SP)*, i. e., the convex hull of all contact points with the ground [8] (plus safety margin), which we compute using *Graham's scan* [32, p.1030]. Although this is strictly speaking not a stability criterion, it describes the feasibility, assuming sufficient friction and the robot being able to realize the corresponding motion. Furthermore, the criterion is only valid if all contact points lie within a horizontal plane, which means climbing stairs or stepping on ramps is not covered. However, in this contribution we consider foot-ground contacts with moderate step heights and ground inclinations, thus we simply project the contact points to the horizontal plane and regard this discrepancy as an additional modeling error.

As in [4], we prefer a slowly moving ZMP to get smoother CoM trajectories. For this purpose, we use linear connections between control points at the start/end of each DS and SS phase. Based on geometric considerations, we use the following scheme to determine the control point positions, cf. Figure 5:

- The first and last control points coincide with the centroids of the corresponding DS-phase SPs.
- The intermediate control points describe the shortest connection between the SPs of two neighboring SS phases. If the shortest connection is not unique, i. e., if the closest edges are parallel, the mean value of the infinite set of shortest connections is used.

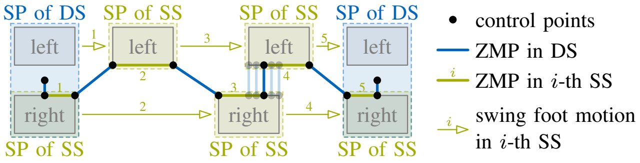


Fig. 5. Proposed strategy for control point placement of the ZMP path.

The main advantages of this method over more elaborate techniques, e. g. the formulation as an optimization problem, as suggested in [4], are its simplicity and the guarantee of finding a valid solution in a deterministic time. Note that the subsequent planning stages are not restricted to linear ZMP paths. In fact, one may use arbitrary shapes, like cubic splines in the DS-phase, as in [14]. The only restriction is $\mathcal{C}^0$-continuity in order to obtain a $\mathcal{C}^2$-smooth CoM motion.

After shaping the path through linear segments, the speed at which the ZMP should traverse this path remains to be specified. For the sake of simplicity, we use constant velocity, which is defined through the segment length and T_{step}. In order to smooth the transition from standing to walking and vice versa, we replace the constant velocity in the first and last segment by a cubic interpolation and used the additional parameters to force zero ZMP velocity and acceleration at the first and last control point. However, this modification is not necessary and yields only slight improvements. Lastly, we point out that the ZMP trajectory is not exclusively used for planning the CoM motion, but also serves as reference input to the SIK module, for details see [27].

Simplified Robot Model / Reduced Model Torso (RMT)

Especially for fast locomotion, one has to ensure that core dynamic effects are reflected by the reduced model used for planning. For this reason, we propose using a three-mass model as suggested in [4], cf. Figure 6 (right), which considers torso and leg dynamics. In contrast to [15], the torso is not modeled as linear inverted pendulum, but instead as free mass m_t which allows us to consider variable heights z_t of the RMT, i. e., the torso center point. Moreover, through the moment of inertia tensor $\mathbf{\Theta}_t$, we incorporate the dynamics of torso rotation. However, in this paper, we stick

PREPRINT - Final published paper: https://doi.org/10.1109/Humanoids43949.2019.9035076

to an upright torso. Note that one could easily incorporate arm dynamics by adding two corresponding point masses, however, we assume arm motion to be a result of null-space optimization in the SIK module, thus arms are not (yet) considered in the WPG module. We plan this extension for our future goal of multi-contact locomotion.

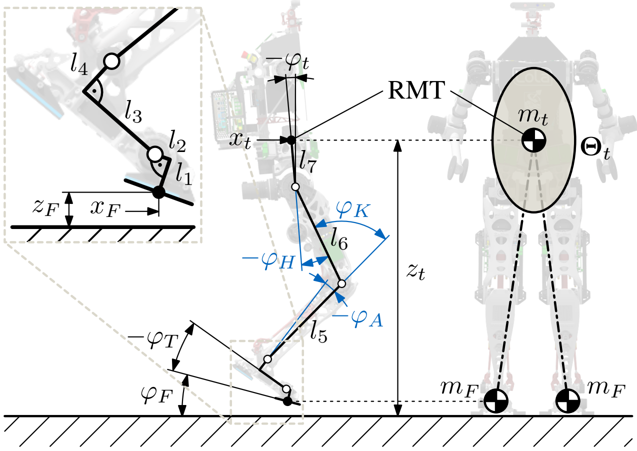


Fig. 6. Left: simplified model of leg kinematics (2D). The foot rotation φ_F and toe-joint angle φ_T are specified in Stage 1, while the torso rotation φ_t is user-defined (here: $\varphi_t = \dot{\varphi}_t = 0$). In contrast, ankle flexion φ_A, knee flexion φ_K, and hip flexion φ_H (blue) are not part of the task-space and thus unknown within the WPG module. All angles are positive clockwise. Right: reduced three-mass model of multi-body dynamics (3D) consisting of point masses for the feet m_F, a point mass m_t coinciding with the RMT (center of torso) and moment of inertia tensor Θ_t of the torso.

Beside the three-mass model which we will use in Stage 4 to compute *horizontal* RMT motion, we introduce a simplified model of leg kinematics, cf. Figure 6 (left), to approximate the kinematic limits of *vertical* RMT motion. While the three-mass model is described fully in 3D *without linearization*, the kinematic model represents a projection of the leg DoFs (cf. Figure 1) to the sagital plane (2D). For walking on curved paths we use additional coordinate systems to deal with three-dimensional rotation of the feet and a changing sagital plane. Here, for the sake of simplicity, we only discuss straight walking. Hence, all expressions are described relative to the inertial reference frame (I-frame), cf. Figure 1. Note that this also holds for Θ_t, which has to be transformed from the torso frame to the I-frame.

Stage 3: Vertical Reduced Model Torso (RMT) Motion

In planning vertical RMT motion, we first specify a desired trajectory $z_{t,\text{des}}(t)$, which can be understood as "ideal" vertical torso motion, neglecting any kinematic limits. Similar to horizontal foot motion, cf. Figure 4, we choose $z_{t,\text{des}}$ to be a piecewise quintic polynomial with stationary control points at the beginning/end of each physical step, i. e., at t_i and t_{i+1} with $\dot{z}_{t,\text{des}}(t_{i|i+1}) = \ddot{z}_{t,\text{des}}(t_{i|i+1}) = 0$. The control points are chosen such that the CoM (as combination of RMT and foot point masses) has constant height above the current stance foot. This way we obtain a reasonable shape which can deal with varying foothold heights, e. g. for climbing stairs. This simple heuristic represents the output of the *first run* of Stage 3 and an initial estimate of $z_t(t) \approx z_{t,\text{des}}(t)$, which we can use in the first run of Stage 4, cf. Figure 3.

For the *second run* of Stage 3, we need knowledge of the horizontal RMT motion (in particular x_t) which in turn depends on z_t. We resolve this circular dependency by running Stage 4 provisionally (with a larger step size, thus lower resolution) between the first and second run of Stage 3, cf. Figure 3. Alternatively, one could use the already known ZMP position as an approximation for $x_t \approx x_{\text{ZMP}}$ and thus avoid running Stage 3 and 4 twice. However, for fast walking, ZMP and horizontal RMT motion display significant differences, which would lead to a poor approximation.

To incorporate kinematic limits in the second run of Stage 3, we use our simplified model, cf. Figure 6 (left). We do this by finding an upper and lower limit $z_{t,\text{max}}$ and $z_{t,\text{min}}$ to which $z_{t,\text{des}}$ is restricted. Note that although the underlying model is almost identical to [6], we use different approaches for both, determining the limits and applying them to $z_t(t)$. Based on [21], the method presented in [6] for example assumes a fully stretched leg, i. e., $\varphi_K = 0$, for estimating $z_{t,\text{max}}$, which is not exact if we consider fixed footholds. In contrast, we evaluate the kinematic chain of the simplified model to compute limits for the torso height given x_F, z_F, φ_F, and φ_T from Stage 1, the user-defined torso rotation φ_t, and the provisional x_t from Stage 4. For this purpose, we introduce the abbreviations $sx := \sin(x)$ and $cx := \cos(x)$. For the torso position x_t, z_t we find

$$\begin{aligned} x_t = x_F &+ l_1\, s\varphi_F - l_2\, c\varphi_F - l_3\, c\alpha + l_4\, s\alpha \\ &+ l_5\, s\beta - l_6\, s\gamma + l_7\, s\varphi_t \ , \end{aligned} \tag{1}$$

$$\begin{aligned} z_t = z_F &+ l_1\, c\varphi_F + l_2\, s\varphi_F + l_3\, s\alpha + l_4\, c\alpha \\ &+ l_5\, c\beta + l_6\, c\gamma + l_7\, c\varphi_t \ , \end{aligned} \tag{2}$$

with $\alpha := \varphi_F - \varphi_T$, $\beta := \alpha - \varphi_A$, and $\gamma := \varphi_K - \beta$. We choose φ_A as free parameter, thus φ_K is the only unknown left to compute z_t. We find φ_K from (1) since x_t is given:

$$\varphi_K = \beta + \sin^{-1} \underbrace{\left(\frac{x_A - x_t}{l_6} + \frac{l_5}{l_6}\, s\beta + \frac{l_7}{l_6}\, s\varphi_t \right)}_{=:\xi} \ . \tag{3}$$

Note that if $|\xi| > 1$ for a given φ_A, there exists no φ_K such that we can reach the desired x_t. Using (3) and (2), we can compute $z_t = f(\varphi_A)$. Moreover, we find $\varphi_H = \beta - \varphi_K - \varphi_t$ which we can use to check the kinematic limits of the hip flexion joint.

Although an analytical solution to the inverse kinematics exists, cf. [6], we could not obtain an analytic expression for $z_{t,\text{max}}$ and $z_{t,\text{min}}$ respecting joint limits for φ_A, φ_K and φ_H. Instead, we use a custom multi-level sampling approach[3] to find the global optima. Note that φ_H and φ_K are implicitly given through φ_A, thus we only have to sample φ_A, which is of moderate cost. We compute $z_{t,\text{max}}$ and $z_{t,\text{min}}$ for both legs, which are then combined in order to respect the kinematic limits of both legs simultaneously.

[3] We also tried gradient-based optimization; however, due to strong nonlinearities, these methods became easily stuck in local optima.

PREPRINT - Final published paper: https://doi.org/10.1109/Humanoids43949.2019.9035076

For the final vertical RMT motion $z_t(t)$, we define equally distributed control points t_k with $\Delta t_k = 0.1\,\mathrm{s}$. We specify these control points by restricting $z_{t,\mathrm{des}}(t_k)$ to the combined limits $z_{t,\max}$ and $z_{t,\min}$, such that $z_{t,\min}(t_k) + z_{t,\mathrm{safety}} \leq z_t(t_k) \leq z_{t,\max}(t_k) - z_{t,\mathrm{safety}}$ holds. Here $z_{t,\mathrm{safety}} \geq 0$ represents a constant safety margin for both boundaries. For complex motions these boundaries may overlap, e. g. for stepping down, cf. Figure 7 at 15.7 s. Although overlapping boundaries indicate that the planned motion is (theoretically) not feasible, it may still be possible in practice, since our simplified model is conservative in neglecting several DoFs of the robot. In such cases we choose the control point to be the mean value of the boundaries, which minimizes the distance to both limits. As a postprocessing step, we limit the gradient between each control point and its successor in order to avoid $\ddot{z}_t > g$, i. e., jumping. This is necessary for motions where $z_{t,\max}$ and $z_{t,\min}$ form a very steep tunnel, e. g. for stepping up, cf. Figure 7 at 10.7 s.

The control points are then interpolated using a $\mathcal{C}^4$-smooth quintic spline respecting initial and end conditions to connect trajectories in a $\mathcal{C}^2$-smooth manner. For interpolation, we use an efficient algorithm proposed in [5]. The resulting vertical RMT motion $z_t(t)$ for stepping up and down a platform of 12.5 cm height is shown in Figure 7. Note that using a simple projection instead of an optimization as in [6] not only significantly reduces computational cost, but we also obtain a smoother trajectory in regions which are "unproblematic", e. g. in Figure 7 at $t \in [11.5 \ldots 14.5\,\mathrm{s}]$.

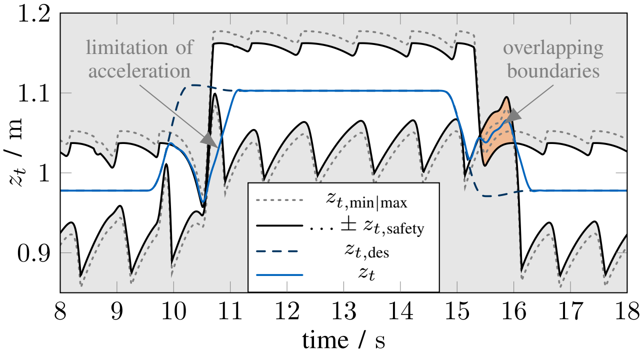


Fig. 7. Planned vertical RMT motion with lower and upper boundaries obtained from simplified kinematic model for stepping up and down a platform of 12.5 cm height. White areas are kinematically feasible, while gray areas denote values above/below the limits. For stepping down, the limits overlap (orange area).

Stage 4: Horizontal Reduced Model Torso (RMT) Motion

Until this point, we have only considered the robot's kinematics. In order to also incorporate dynamic effects, we use our three-mass model, cf. Figure 6 (right). We obtain the (3D) nonlinear rotational *Equations of Motion (EoM)* as

$$\begin{aligned} m_t\,(\boldsymbol{r}_t \times (\boldsymbol{g} - \ddot{\boldsymbol{r}}_t)) = \boldsymbol{\omega}_t \times (\boldsymbol{\Theta}_t\,\boldsymbol{\omega}_t) + \boldsymbol{\Theta}_t\,\dot{\boldsymbol{\omega}}_t - \boldsymbol{T}_{\mathrm{cont}} \\ + m_F\,\boldsymbol{r}_{F,\mathrm{right}} \times (\ddot{\boldsymbol{r}}_{F,\mathrm{right}} - \boldsymbol{g}) \\ + m_F\,\boldsymbol{r}_{F,\mathrm{left}} \times (\ddot{\boldsymbol{r}}_{F,\mathrm{left}} - \boldsymbol{g}) \end{aligned} \tag{4}$$

where $\boldsymbol{g} = [0,\,0,\,-g]^T$ denotes the gravity vector and $\boldsymbol{r}_t$, $\boldsymbol{r}_{F,\mathrm{right}}$, $\boldsymbol{r}_{F,\mathrm{left}}$ the position of the torso (RMT), right and left foot, respectively. Moreover, $\boldsymbol{\omega}_t$ and $\dot{\boldsymbol{\omega}}_t$ denote the angular velocity and acceleration of the torso (user-defined; here: $\boldsymbol{\omega}_t = \dot{\boldsymbol{\omega}}_t = \boldsymbol{0}$), which together with $\boldsymbol{\Theta}_t$ account for torso rotation. Lastly, $\boldsymbol{T}_{\mathrm{cont}}$ is the contact moment, where the first two components are given by the ZMP (Stage 2) with

$$T_{\mathrm{cont},1} = y_{\mathrm{ZMP}}\,m\,(\ddot{z}_{\mathrm{CoM}} + g),\ T_{\mathrm{cont},2} = -x_{\mathrm{ZMP}}\,m\,(\ddot{z}_{\mathrm{CoM}} + g)\ .$$

Here $m = m_t + 2\,m_F$ is the total mass and $\ddot{z}_{\mathrm{CoM}}$ is given by

$$\ddot{z}_{\mathrm{CoM}} = \frac{1}{m}\,(m_t\,\ddot{z}_t + m_F\,\ddot{z}_{F,\mathrm{right}} + m_F\,\ddot{z}_{F,\mathrm{left}})\ .$$

Note that $\ddot{z}_t$ is available from Stage 3, thus the first two rows of the *Right-Hand Side (RHS)* of (4) are known. If we expand the left-hand side, we obtain for the first two rows

$$\begin{aligned} m_t\,z_t\,\ddot{y}_t - m_t\,(\ddot{z}_t + g)\,y_t &= \mathrm{RHS}_1\ (\mathrm{known})\ , \\ -m_t\,z_t\,\ddot{x}_t + m_t\,(\ddot{z}_t + g)\,x_t &= \mathrm{RHS}_2\ (\mathrm{known}) \end{aligned} \tag{5}$$

which represent two *decoupled* linear ordinary differential equations for $x_t(t)$ and $y_t(t)$. Within the WPG module, we typically plan the motion from a given initial pose to a given end pose. These poses represent boundary conditions

$$\begin{aligned} \boldsymbol{r}_t(t_0) &= \boldsymbol{r}_{t,0}, & \dot{\boldsymbol{r}}_t(t_0) &= \dot{\boldsymbol{r}}_{t,0}, & \ddot{\boldsymbol{r}}_t(t_0) &= \ddot{\boldsymbol{r}}_{t,0}\ , \\ \boldsymbol{r}_t(t_{\mathrm{end}}) &= \boldsymbol{r}_{t,\mathrm{end}}, & \dot{\boldsymbol{r}}_t(t_{\mathrm{end}}) &= \dot{\boldsymbol{r}}_{t,\mathrm{end}}, & \ddot{\boldsymbol{r}}_t(t_{\mathrm{end}}) &= \ddot{\boldsymbol{r}}_{t,\mathrm{end}} \end{aligned} \tag{6}$$

which together with (5) represent two independent second-order linear BVPs. Note that the BVPs are *overdetermined*; thus, in general, the solution *does not exist*. In [4], a collocation method using cubic splines has been proposed to obtain a reasonable approximation of $x_t(t)$ and $y_t(t)$, even if the "real" solution does not exist. This is done by generating a $\mathcal{C}^2$-smooth cubic spline such that (4) is fulfilled at equally distributed collocation sites, while simultaneously satisfying the boundary conditions (6). In order to fulfill all components of (6), $\boldsymbol{T}_{\mathrm{cont}}$ had to be modified by manually tuned trapezoidal shape functions and virtual control points. In contrast to [4], we propose collocation based on *quintic* splines, which can by construction include boundary conditions up to the second-order derivative, thus removing the need for modifying the EoMs or adding virtual control points. This algorithm is a combination and extension of [4], [31], [33] and is derived in detail in [5], where we also show that, for sufficiently smooth ODEs, using quintic splines is in general more efficient than the cubic counterpart. In particular, a lower count of (expensive) collocation sites leads to similar results.

As explained in [5], it is difficult to quantify the approximation quality. This is due to the fact that we try to approximate a solution which actually does not exist. Thus, it is not possible to formulate an error. However, as shown in [5], the root mean square of the residual of (4) can be used to detect divergence. For both cubic and quintic spline collocation, we find $\Delta t_{\mathrm{coll}} = 0.2\,\mathrm{s}$ for the *first run* and $\Delta t_{\mathrm{coll}} = 0.1\,\mathrm{s}$ for the *second run* of Stage 4 to be an appropriate distance between collocation sites. We would like to highlight that the algorithms used for cubic/quintic spline interpolation/collocation are available as part of our free and open-source C++ library BROCCOLI [34].

Stage 5: Center of Mass Motion

After planning trajectories for the feet and torso, it is straightforward to obtain the CoM motion from our three-mass model with

$$\boldsymbol{r}_{\text{CoM}}(t) = \frac{1}{m}\left(m_t\,\boldsymbol{r}_t(t) + m_F\,\boldsymbol{r}_{F,\text{right}}(t) + m_F\,\boldsymbol{r}_{F,\text{left}}(t)\right) \ .$$

Note that $\boldsymbol{r}_t(t)$ is $\mathcal{C}^4$-smooth due to quintic spline collocation; however, we constrained $\boldsymbol{r}_F(t)$ only to be $\mathcal{C}^2$-smooth. Thus, $\boldsymbol{r}_{\text{CoM}}(t)$ is $\mathcal{C}^4$-smooth in most parts, but only $\mathcal{C}^2$-smooth at the control points of foot motion. However, we still obtain a significant reduction of motor jerk over the previous implementation with cubic splines, cf. Figure 8 (right). We see this improvement as main argument for using quintic splines, thus bypassing our previous dilemma of not being able to quantify approximation quality.

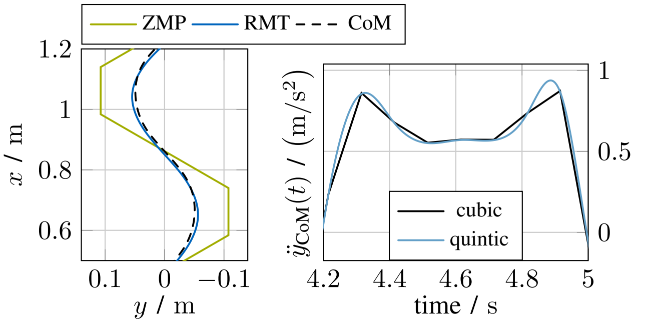


Fig. 8. Left: planned paths for ZMP, RMT and CoM obtained from proposed method. Right: comparison of lateral CoM acceleration for cubic ($\mathcal{C}^2$-smooth) and quintic ($\mathcal{C}^4$-smooth) spline collocation.

V. Results

For evaluation, we chose a scenario where the robot is commanded to step up and down a platform of 12.5 cm height, cf. Figure 9. We used a predefined sequence of footholds (step-count: 20, duration of motion: 17.6 s) and triggered Stages 1-5 from Section IV for which we measure the following runtimes:

Stage	1	2	3.1	3.2	4.1	4.2	5
Runtime [ms]	0.251	0.203	0.027	1.309	1.245	5.221	0.214

So as to react dynamically to changes of the environment and user-input, our goal is to trigger the WPG module at least every $T_{\text{step,min}} = 0.6\,\text{s}$. With a total runtime of 8.5 ms we easily satisfy this requirement, even for a planning horizon of more than 17 s. We highlight that, in contrast to the previous method from [4] and [6], where two steps are planned ahead continuously, our new implementation plans the complete sequence consisting of 20 steps at the same time. By doing so, we avoid planning a periodic reference as boundary condition as in [4]. Since we avoid expensive optimization, planning the complete sequence with our new approach is approximately 10 times faster than the previous method.

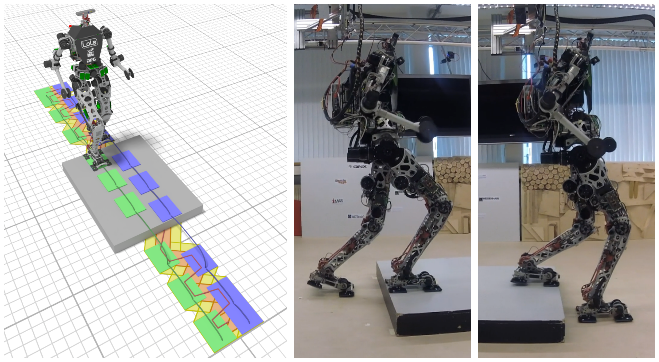


Fig. 9. Chosen scenario for evaluation. Left: planned motion over platform in simulation environment. Right: snapshots of stepping up and down during experiment.

We successfully conducted simulations[4] and experiments, which are best presented as a video (available online under https://youtu.be/piQm_oTYXIc). Note that the chosen test case is the same setup as in [6], however, with a different foothold sequence. As a qualitative measure for stability, we observe reduced arm motion in simulations and experiments, which indicates that it is "easier" for the SIK module to follow the planned CoM trajectory (less action in null-space required). To quantify stability in experiments, we compare the error of torso rotation $\Delta\varphi_t$, which could be significantly reduced, cf. Figure 10.

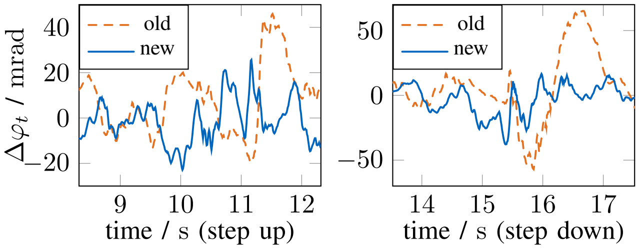


Fig. 10. Experimental data for error of torso rotation $\Delta\varphi_t$ (measured by IMU) as indicator for stability. Comparison of previous ("old") and proposed ("new") method for stepping up (left) and stepping down (right). Achieved reduction of $\max|\Delta\varphi_t|$: 44 % (step up) and 42 % (step down).

Moreover, using the proposed method, we were able to increase the distance to kinematic limits of the "critical" knee flexion and ankle flexion joints, cf. Figure 11.

For the sake of completeness, we give a list of relevant parameters which we used to obtain the results: $L_{\text{foot}} = 0.276\,\text{m}$, $L_{\text{step},i} = 0\ldots0.7\,\text{m}$, $\Delta z_{\text{step}} = 0.03\,\text{m}$, $T_{\text{step},i} = 0.8\ldots1\,\text{s}$, $m_t = 54\,\text{kg}$, $m_F = 3\,\text{kg}$, $g = 9.81\,\text{m/s}^2$.

VI. Conclusions and Future Work

The proposed method allows us to 1) reduce algorithmic complexity, 2) significantly reduce execution time ($-90\,\%$), 3) increase stability ($+40\,\%$), and 4) increase distance to kinematic limits ($+50\,\%$) during challenging maneuvers. Moreover, the presented algorithm is very general, which makes it easy to develop modifications or extensions.

[4] For simulation, we use a custom environment, cf. [31].

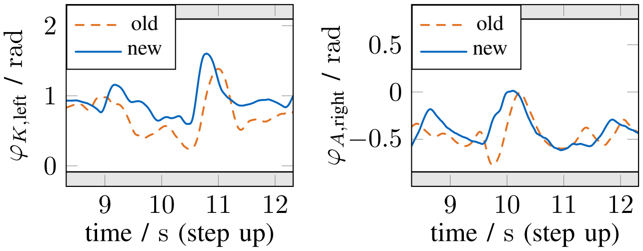


Fig. 11. Experimental data for left knee flexion $\varphi_{K,\text{left}}$ (left) and right ankle flexion $\varphi_{A,\text{right}}$ (right) as kinematically "critical" joints for stepping up. Comparison of previous ("old") and proposed ("new") method. Gray areas represent kinematic limits. Achieved increase of kinematic "reserves": $\min|\varphi_K - \varphi_{K,\text{limit}}| \to +54\,\%$ and $\min|\varphi_A - \varphi_{A,\text{limit}}| \to +222\,\%$.

For our future goal of multi-contact locomotion, we plan to consider different formulations of balance, like pure static criteria as in [35], extensions of the ZMP to non co-planar contacts as in [36] or interpreting additional contact forces as modification of the support polygon as in [37]. Staying faithful to our strategy so far, we plan to maintain robustness and real-time performance also in future.

Acknowledgment

This work is supported by the *German Research Foundation* (DFG, project number 407378162).

References

[1] C. G. Atkeson *et al.*, "No Falls, No Resets: Reliable Humanoid Behavior in the DARPA Robotics Challenge," in *IEEE-RAS 15th International Conference on Humanoid Robots*, 2015.

[2] G. Nelson, A. Saunders, and R. Playter, "The PETMAN and Atlas Robots at Boston Dynamics," in *Humanoid Robotics: A Reference*, A. Goswami and P. Vadakkepat, Eds. Springer Netherlands, 2019.

[3] Y. Sakagami *et al.*, "The intelligent ASIMO: System overview and integration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.

[4] T. Buschmann *et al.*, "A Collocation Method for Real-Time Walking Pattern Generation," in *IEEE-RAS International Conference on Humanoid Robots*, 2007.

[5] P. Seiwald and D. Rixen, "Fast Approximation of Over-Determined Second-Order Linear Boundary Value Problems by Cubic and Quintic Spline Collocation," *International Journal for Numerical Methods in Engineering*, Preprint – submitted. [Online]. Available: https://mediatum.ub.tum.de/1509742.

[6] A.-C. Hildebrandt *et al.*, "Torso height optimization for bipedal locomotion," *International Journal of Advanced Robotic Systems*, vol. 15, no. 5, 2018.

[7] M. Vukobratović and J. Stepanenko, "On The Stability of Anthropomorphic Systems," *Mathematical Biosciences*, vol. 15, no. 1, 1972.

[8] M. Vukobratović and B. Borovac, "Zero-Moment Point – Thirty Five Years of its Life," *International Journal of Humanoid Robotics*, vol. 1, no. 1, 2004.

[9] K. Nagasaka, H. Inoue, and M. Inaba, "Dynamic Walking Pattern Generation for a Humanoid Robot Based on Optimal Gradient Method," in *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 6, 1999.

[10] K. Nishiwaki *et al.*, "Online Generation of Humanoid Walking Motion based on a Fast Generation Method of Motion Pattern that Follows Desired ZMP," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, 2002.

[11] S. Kajita and K. Tani, "Experimental Study of Biped Dynamic Walking in the Linear Inverted Pendulum Mode," in *IEEE International Conference on Robotics and Automation*, vol. 3, 1995.

[12] ——, "Study of Dynamic Biped Locomotion on Rugged Terrain – Derivation and Application of the Linear Inverted Pendulum Mode," in *IEEE International Conference on Robotics and Automation*, 1991.

[13] S. Kajita *et al.*, "The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern generation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001.

[14] S. Kajita *et al.*, "Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point," in *IEEE International Conference on Robotics and Automation*, vol. 2, 2003.

[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*, 2009.

[16] J. Englsberger, C. Ott, and A. Albu-Schäffer, "Three-dimensional bipedal walking control using Divergent Component of Motion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013.

[17] ——, "Three-Dimensional Bipedal Walking Control Based on Divergent Component of Motion," *IEEE Transactions on Robotics*, vol. 31, no. 2, 2015.

[18] J. Pratt *et al.*, "Capture Point: A Step toward Humanoid Push Recovery," in *IEEE-RAS International Conference on Humanoid Robots*, 2006.

[19] J. Englsberger *et al.*, "Bipedal walking control based on Capture Point dynamics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[20] O. Stasse *et al.*, "Strategies for Humanoid Robots to Dynamically Walk Over Large Obstacles," *IEEE Transactions on Robotics*, vol. 25, no. 4, 2009.

[21] K. Nishiwaki and S. Kagami, "Online Design of Torso Height Trajectories for Walking Patterns that takes Future Kinematic Limits into Consideration," in *IEEE International Conference on Robotics and Automation*, 2011.

[22] S. Lengagne *et al.*, "Generation of whole-body optimal dynamic multi-contact motions," *The International Journal of Robotics Research*, vol. 32, no. 9–10, 2013.

[23] S. Lohmeier, "Design and Realization of a Humanoid Robot for Fast and Autonomous Bipedal Locomotion," Dissertation, Technical University of Munich, 2010. [Online]. Available: https://mediatum.ub.tum.de/980754.

[24] F. Sygulla *et al.*, "An EtherCAT-Based Real-Time Control System Architecture for Humanoid Robots," in *IEEE 14th International Conference on Automation Science and Engineering*, 2018.

[25] K. Shoemake, "Animating Rotation with Quaternion Curves," *SIGGRAPH Computer Graphics*, vol. 19, no. 3, 1985.

[26] F. Sygulla *et al.*, "Hybrid Position/Force Control for Biped Robot Stabilization with Integrated Center of Mass Dynamics," in *IEEE-RAS International Conference on Humanoid Robots*, 2017.

[27] F. Sygulla and D. Rixen, "A Force Control Scheme for Biped Robots to Walk over Uneven Terrain Including Partial Footholds," *Preprint - submitted*, 2019, DOI: 10.13140/RG.2.2.19012.68487.

[28] A. Liégeois, "Automatic Supervisory Control of the Configuration and Behaviour of Multibody Mechanisms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 7, no. 12, 1977.

[29] M. Schwienbacher *et al.*, "Self-collision avoidance and angular momentum compensation for a biped humanoid robot," in *IEEE International Conference on Robotics and Automation*, 2011.

[30] A.-C. Hildebrandt *et al.*, "Real-Time Path Planning in Unknown Environments for Bipedal Robots," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, 2017.

[31] T. Buschmann, "Simulation and Control of Biped Walking Robots," Dissertation, Technical University of Munich, 2010. [Online]. Available: https://mediatum.ub.tum.de/997204.

[32] T. H. Cormen *et al.*, *Introduction to Algorithms*, 3rd ed. The MIT Press, 2009.

[33] E. H. Mund, P. Hallet, and J. P. Hennart, "An algorithm for the interpolation of functions using quintic splines," *Journal of Computational and Applied Mathematics*, vol. 1, no. 4, 1975.

[34] P. Seiwald and F. Sygulla, *broccoli: Beautiful Robot C++ Code Library*, 2019. [Online]. Available: https://gitlab.lrz.de/AM/broccoli.

[35] T. Bretl and S. Lall, "Testing Static Equilibrium for Legged Robots," *IEEE Transactions on Robotics*, vol. 24, no. 4, 2008.

[36] S. Caron, Q.-C. Pham, and Y. Nakamura, "ZMP Support Areas for Multicontact Mobility Under Frictional Constraints," *IEEE Transactions on Robotics*, vol. 33, no. 1, 2017.

[37] S. Mason *et al.*, "An MPC Walking Framework With External Contact Forces," in *IEEE International Conference on Robotics and Automation*, 2018.