Research Article

INTERNATIONAL JOURNAL OF ADVANCED ROBOTIC SYSTEMS

International Journal of Advanced Robotic Systems
September-October 2018: 1–11

DOI: 10.1177/1729881418804442
journals.sagepub.com/home/arx
SAGE

Torso height optimization for bipedal locomotion

Arne-Christoph Hildebrandt, Konstantin Ritt, Daniel Wahrmann, Robert Wittmann, Felix Sygulla, Philipp Seiwald, Daniel Rixen and Thomas Buschmann

Abstract

Bipedal robots can be better alternatives to other robots in certain applications, but their full potential can only be used if their entire kinematic range is cleverly exploited. Generating motions that are not only dynamically feasible but also take into account the kinematic limits as well as collisions in real time is one of the main challenges towards that goal. We present an approach to generate adaptable torso height trajectories to exploit the full kinematic range in bipedal locomotion. A simplified 2D model approximates the robot's full kinematic model for multiple steps ahead. It is used to optimize the torso height trajectories while taking future motion kinematics into account. The method significantly improves the robot's motion not only while walking in uneven terrain, but also during normal walking. Furthermore, we integrated the method in our framework for autonomous walking and we validated its real-time character in successfully conducted experiments.

Keywords

Biped robots, locomotion, walking control, climbing stairs

Date received: 12 April 2018; accepted: 2 September 2018

Topic: Climbing and Walking Robots
Topic Editor: Yannick Aoustin
Associate Editor: Antoni Świć

Introduction

The capability to step up and down platforms or stairs is the main advantage of legged over wheeled locomotion. Bipedal robots show impressive results in terms of walking and running in even terrain.[1,2] But legged locomotion allows for much more complex movement than just following a continuous path.[3] Motion generation for stepping up and down is directly coupled with the open question of how the height of the robot's center of mass with respect to the height of the torso has to be designed. A variable torso height can result in several advantages for bipedal walking. Humans make great use of the advantage of a variable torso height during walking. Imitating a human-like walking may improve the energy efficiency.[4] Adapting the torso height to the current walking situation and terrain yields a greater kinematic versatility: larger strides are possible[5] and the maneuverability on stairs and on uneven terrain is improved.[3] Kinematic constraints such as joint limits can be avoided. Furthermore, recent publications have even showed a self-stabilization influence by applying a well-designed torso height trajectory.[6,7]

In this article, we focus on methods to improve the kinematic versatility of bipedal walking by tightly integrating a new torso height trajectory in our framework for bipedal locomotion. Furthermore, we present an optimization technique to reduce joint velocities and to avoid

Department of Mechanical Engineering, Technical University of Munich (TUM), Munich, Germany

Corresponding authors:
Arne-Christoph Hildebrandt and Konstantin Ritt, Department of Mechanical Engineering, Technical University of Munich (TUM), Boltzmannstr 15, 85748 Garching, Germany.
Emails: arne.hildebrandt@mytum.de; konsiritt@gmail.com

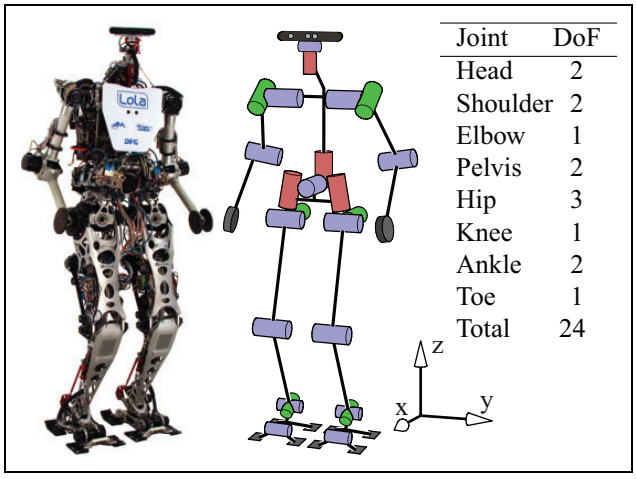

Figure 1. Photo and kinematic structure of the humanoid robot *Lola*. The joint distribution and world coordinate system used are shown on the right side (adapted from the work of Hildebrandt et al.[8]).

reaching kinematic constraints. In the next section, we present related works and highlight the originality of our contribution. The following section provides an overview of the experimental platform used in this work—the robot *Lola* (Figure 1)—and its framework for real-time motion generation. Then, we present our new torso height trajectory and its optimization. Finally, the new trajectory optimization method is evaluated in simulations and validated in successfully conducted experiments. The article concludes with a conclusion and comments on future work.

Literature review

Motion generation for humanoid robots often follows a hierarchical approach.[1–3,9] A navigation module determines desired foothold positions depending on the environment and user input. Based on the foothold positions, a walking pattern generation calculates a set of reference trajectories using simplified models which approximate the robot's multibody system. Usually, these trajectories include the center of mass (CoM) trajectories to allow the robot for dynamically feasible walking. Over the last decades, a large variety of models and methods have been presented to derive dynamically feasible CoM trajectories in the literature[1–3,9], among others. Often the robot is represented by an inverted pendulum where the height is considered as constant. Variable CoM height in motion generation is taken into account in the literature,[1,10,11] among others. Few approaches take into account kinematic and dynamic feasibility constraints in an integrated manner.[12] Brasseur et al. formulate linear constraints that guarantee dynamic feasibility and allow efficient solving and, thus, achieving online generation of 3-D trajectories. Nevertheless, their approach was only applied in simulation and not integrated in a whole framework for motion generation in cluttered environments. However, in humanoid robotics, vertical CoM trajectories are often generated in a heuristic manner and without considering their influence on the overall motion.

The authors of previous studies,[5,6,13] for example, propose to use a cosine oscillation around an average height. The choice is motivated by observation of human walking. All works analyze the effects of vertical oscillations only in simulation.

To allow climbing of stairs, Park et al.[14] use a sixth-order polynomial to generate the CoM height trajectory. They define boundary conditions and design parameters that are tuned in simulation to reduce the deviation of the real from the ideal planned zero moment point (ZMP). This yields an almost linear function, which the authors used for simplicity. Drawbacks of this simplification are not examined.

Hong and Lee[15] deploy varying trajectories in dual and single stance. While a constant CoM height is assumed in single stance, cubic splines are used for the double support phase. This interpolation allows a transition to different CoM heights. The authors do not provide a rationale to motivate their approach, but they demonstrate in simulations the ability to walk stably in uneven terrain.

Miura et al.[16] always set the robot's waist height as high as possible. When the legs are fully stretched and therefore reach their limit, the waist height is lowered. The resultant trajectory suffers from non-smooth transitions between double and single stance phase. The authors therefore smooth the trajectory in an optimization with respect to a cost function constraining joint angles and velocities. The objective of the publication was to accurately imitate human-like motion, which was demonstrated on even terrain.

Similar to Miura et al., Griffin et al.[17] favor walking with straightened legs over bent legs in their approach. Instead of planning torso height trajectories directly, they propose to project leg joint angle objectives into the null-space of the quadratic optimization problem which is solved by a whole-body controller. The resulting deviation of the CoM height from the reference trajectory introduce additional perturbations which have to be handled by the walking controller. Trying to imitate the human walking, Hu et al.[18] generate the CoM height motion indirectly by introducing angle objectives in the quadratic optimization program which is solved by their motion controller as well. Both publications[17,18] do not take into account future motion limits. They consider only the current controller time step, and they do not optimize over a motion sequence as, for example, a physical step of the robot. Nishiwaki[10] proposed a similar approach as well. Keeping the torso height as high as possible while taking into account the solvability of the inverse kinematics and joint limits. First, they calculate the maximum feasible torso height based on the maximum joint velocities and maximum kinematically adjusted torso height. Second, the torso trajectory is designed iteratively. In difference to the previous publications, they take into account future upper limit of waist height, vertical velocity, and acceleration limits. The torso

height is then set as high as possible within these constraints, neglecting lower limits of the waist height.

In the study by Nishiwaki et al.,[3] the approach was modified using cubic splines which are defined by three heuristically defined support points per step. These points are chosen in such a way that the torso height stays close to its maximum, avoiding knee singularities.

Recently, optimization frameworks have been presented which generate the robots motion for long walking sequences while taking dynamics as well as kinematic constraints into account.[19,20] These approaches are either pure off-line methods[20] or still need initially long calculation times.[19]

Our approach differs from the previous ones. We present a new parametrization for the torso height based on a cubic spline representation. It can be configured by a variable number of control points. This trajectory representation allows us to extend the set of parameters used in our method for model-predictive kinematic optimization.[21] Using the strategy presented by Buschmann et al.,[22] the torso height is explicitly taken into account to generate dynamically feasible CoM motions.

In contrast to the study by Nishiwaki,[10] not only constraints are avoided in that way but also the joint velocities are optimized.

Furthermore, we target a real-time solution. Our robot should walk with normal human walking speed, and it should be able to react to changing user commands or changing environment within one step. For this reason, the optimization of the large number of free parameters cannot be performed using the full kinematic model. Therefore, we introduce a simplified model approximating the kinematics. This model is used to separately solve a parameter optimization for the torso height trajectory with respect to an objective function, taking into account kinematic limitations and joint velocities. The model is verified and the optimization is analyzed in simulations and validated in experiments.

Control architecture

We use our humanoid robot *Lola* for the method's validation. Figure 1 depicts the robot and its kinematic structure. The following sections provide a short overview of the currently implemented control system needed to understand the integration of the proposed method in our framework for autonomous walking. More details on the mechanical design and the control architecture are given by Buschmann et al.[2]

Control architecture

An overview of our hierarchical walking control system is depicted in Figure 2. The *Vision System*[23] uses only the onboard camera and approximates the environment with swept sphere volumes (SSVs). The representation of the environment and of the robot via SSV objects is used consistently in all control modules for fast distance calculations.

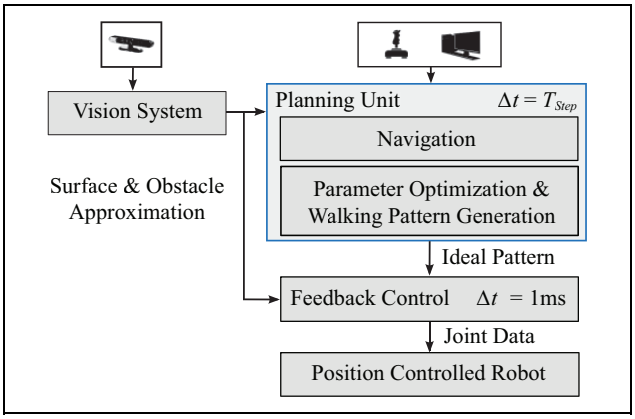

Figure 2. *Lola*'s real-time walking control system.

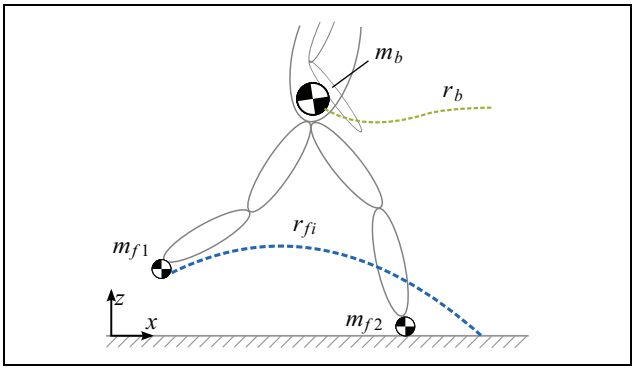

Figure 3. Three-mass model used for CoM trajectory calculation.[24] CoM: center of mass.

The robot control is divided into a high-level *Planning Unit* which is executed for every walking step k (step time of $T_{\text{Step}} = 0.6..1.2\,\text{s}$) and a *Feedback Control* with a cycle time of 1 ms. The *Planning Unit* first calculates, in the *Navigation* module,[8] a sequence of parameter sets $\boldsymbol{p}_{wp,k}$ which configures the robot's walking pattern for the next n_{steps} based on user input, such as desired step parameters or a goal position. These parameter sets determine the overall motion of the robot and include the foothold positions, the final position of the torso height, the step time, and the foot trajectories. Based on $\boldsymbol{p}_{wp,k}$ the *Parameter Optimization & Walking Pattern Generation* evaluates and optimizes $\boldsymbol{p}_{wp,1}$ and generates an ideal walking pattern. The ideal walking pattern serves as input to the *Feedback Control* that adapts the ideal walking pattern according to sensor feedback and calculates joint target data which are executed by the robot. The generation of the CoM trajectory and the parameter optimization will be detailed next.

CoM trajectory generation

The horizontal CoM trajectories are generated as part of the walking pattern generation using a three-mass model to account for dynamic effects caused by fast leg movements (see Figure 3). It has one lumped mass m_b representing the

upper body and masses m_{fi} $(i = 1, 2)$ approximating the leg's dynamics. The input to the horizontal CoM trajectory generation are the trajectories defining the motion of the three masses: foot trajectories $\boldsymbol{r}_{fi}(\boldsymbol{p}_{wp}, t) = [x_{fi}(\boldsymbol{p}_{wp}, t), y_{fi}(\boldsymbol{p}_{wp}, t), z_{fi}(\boldsymbol{p}_{wp}, t)]$, the desired torso height trajectory $z_b = z_b(\boldsymbol{p}_{wp}, t)$ describing the movement of the upper body mass m_b and the desired contact moments trajectories T_x and T_y. The torso height $z_b = z_b(\boldsymbol{p}_{wp}, t) = z_b(H, t)$ is a fifth-order polynomial with a configurable height H at the end of each step (For the sake of simplicity, we will omit the explicit dependency on $\boldsymbol{p}_{wp}$ in the following.).

Thus, the equation of motion (EOM) results in a linear and time variant differential equation for the horizontal upper body mass trajectories x_b and y_b. The equation for the frontal plane can be stated as

$$m_b z_b \ddot{y}_b - m_b y_b(\ddot{z}_b + g) = -T_x + m_f y_{f1}(\ddot{z}_{f1} + g) - m_f z_{f1} \ddot{y}_{f1} + m_f y_{f2}(\ddot{z}_{f2} + g) - m_f z_{f2} \ddot{y}_{f2} \quad (1)$$

The equation for the sagittal plane can be derived analogously. We use the method based on spline collocation proposed in the study by Buschmann et al.[22] to solve for x_b and y_b over two steps.

The *Feedback Control* does not track the desired trajectory of the upper body but of the CoM.[22] The overall CoM trajectories CoM can be calculated by superposition of the motions of the masses m_b and m_{fi}.

Kinematic evaluation and parameter optimization

The parameter set $\boldsymbol{p}_{wp}$ configures the walking pattern of the robot and governs the robot's motion. Off-line–defined parameters are not optimal with respect to dynamic and complex scenarios. Parameters that are chosen without knowledge about the future movement of the robot could lead to walking patterns that are not dynamically feasible or kinematically executable. Therefore, in our previous work,[21] we introduced the *Parameter Optimization* (see Figure 2). It is a model-predictive approach, which uses the full kinematic model of the robot and takes whole-body collision avoidance into account. Based on the results of the model prediction, $\boldsymbol{p}_{wp}$ is evaluated and optimized in real time.

Limitations of the current approach. Currently z_b is generated as a quintic polynomial interpolation between the current height and a control point at the end of the step. The height of this control point is described by the parameter H. The *Parameter Optimization* directly influences z_b by calculating an optimal control point value H. The parameterization with only one parameter per step limits the possibilities to influence the overall motion. It may result in violation of kinematic constraints in challenging scenarios: in fact, joint limits define a maximum or minimum feasible torso height, which in turn restrict indirectly the CoM height. Or, expressed the other way around, a wrongly chosen torso or CoM height trajectory may result in joint limit violations. The minimum feasible torso height is commonly reached when the ankle joint cannot be further flexed. The maximum feasible torso height is mostly determined by the knee joint limits. A typical scenario in which the current trajectory design reaches its limits is stepping up and down stairs or platforms (a video showing *Lola* stepping up and down a platform with the current trajectory design is available at https://youtu.be/rKsx8HKvBkg).

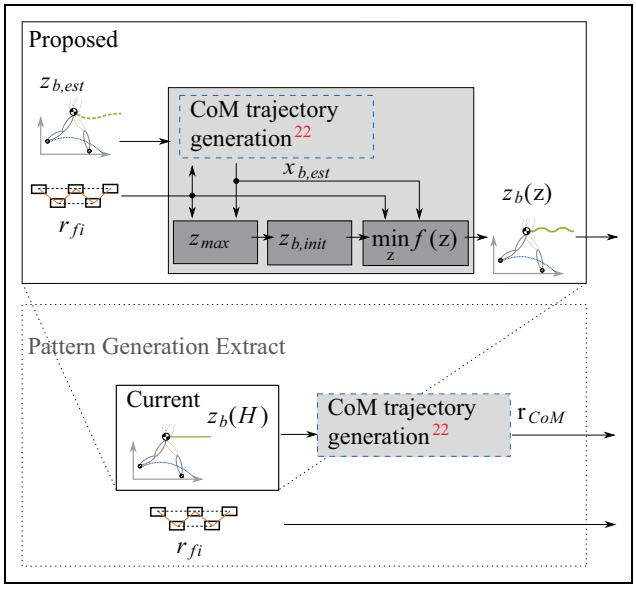

Figure 4. Overview of the proposed method in the context of the pattern generation process. This figure focuses on the optimization process of the torso height.

The *Parameter Optimization* uses the complex full-kinematic model of the robot to analyze the stepping motion of the robot's next physical step. Due to the time-consuming integration of the complex model, it is not possible to introduce additional parameters without violating real-time constraints. For the same reason, it is not possible to analyze more than one walking step in advance. This can become a limiting factor for walking over challenging terrain.

Proposed method

This approach extends the *Parameter Optimization* with a reduced kinematic model that allows us for a more sophisticated torso height trajectory design and longer time horizons.

Figure 4 gives an overview of how the proposed method extends the current pattern generation procedure. To maintain the real-time capability, we divide the trajectory generation into two parts. First, we separately calculate an optimized torso height trajectory, as described in this section. The new torso height trajectory parametrization can

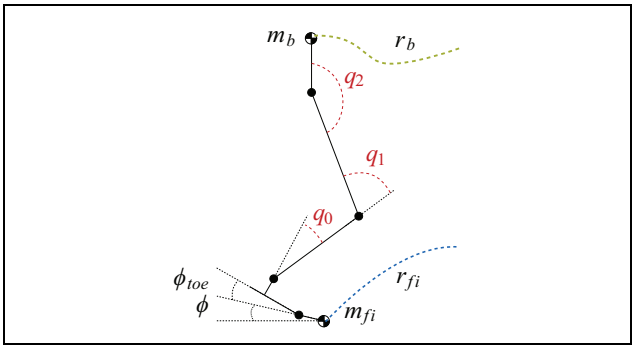

Figure 5. 2-D kinematic model approximating robot's full kinematics. Input trajectories: $\boldsymbol{r}_b$, $\boldsymbol{r}_{fi}$, ϕ, and ϕ_{toe}. Joint angles in red are unknown.

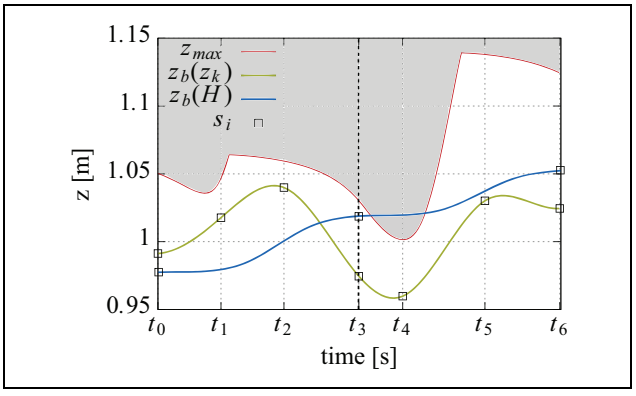

Figure 6. Walking up a platform with $\Delta z_{\text{stair}} = 7.5$ cm: control points of torso height trajectories for two walking steps (separated by dotted line). Blue line denotes trajectory resulting from optimization of H, green line denotes trajectory resulting from presented method.

be configured with more than one parameter per step. We use a simplified kinematic model which is introduced in the following. This model is then employed to obtain an initial solution and to finally perform the optimization. Second, we use the resulting optimized torso height trajectory in the *Parameter Optimization* to determine the remaining parameters describing the robot's motion as presented in our previous study.[21] This is a trade-off between optimality of the solution and calculation time for real-time application.

Parameterization of torso height trajectory

Currently, the torso height trajectory is composed of quintic polynomials with two control points at the start and end of each step (see Figure 6). This representation provides C^2-smooth trajectories, which is important in order to avoid undesired jumps on acceleration level. A step describes the interval between two consecutive double support phases. Using additional control points per step would allow for more variable trajectories. The drawback of the representation with a quintic polynomial is that the first and second derivative have to be set at each control point. Consequently, each set point introduces three new degrees of freedom (DoFs), increasing the dimension of the optimization problem. Furthermore, higher order polynomials tend to introduce undesired oscillations. Instead, we choose to represent the torso height trajectory using cubic splines. Each additional control point introduces 1 DoF to the curve representation, while keeping the spline property of C^2-smoothness. The new torso height trajectory is represented by four control points per step with the heights $z_k, k = 0 \ldots 3$. Initial conditions reduce the DoFs to three parameters per step. We do not impose further boundary constraints. In the following sections, the choice for the control points is discussed in more detail.

Simplified kinematic model

During the optimization, the underlying model is evaluated frequently. To maintain real-time capability, a low complexity of the model is desirable. The robot's kinematics are approximated with a simple kinematic 2-D model in the sagittal plane. Thus, the 2-D kinematic chain of each leg is fully determined with the trajectories $\boldsymbol{r}_b = [x_b, y_b, z_b]$, $\boldsymbol{r}_{fi}$, ϕ, and the heuristically defined toe trajectory ϕ_{toe}. Figure 5 shows a 2-D sketch of one leg depicting the variables. In contrast to the full model used in the *Parameter Optimization*, collision checks are not considered. The trajectories of $\boldsymbol{r}_{fi}$, ϕ, and ϕ_{toe} are defined by $\boldsymbol{p}_{wp}$. The horizontal trajectory of the body mass point of the three-mass model x_b and y_b results from solving the governing EoMs (see Figure 1). As explained in the previous section, the torso height trajectory z_b itself is a necessary input to solve these equations. An estimated torso height trajectory $z_{b,\text{est}}$ is used to solve the EoMs to yield a sufficiently accurate horizontal trajectory $x_{b,\text{est}}$. When $\boldsymbol{r}_b$, $\boldsymbol{r}_{fi}$, ϕ, and ϕ_{toe} are determined, the inverse kinematics can be solved analytically for the joint angles $\boldsymbol{q}$. The joint velocities $\dot{\boldsymbol{q}}$ are derived numerically. Thus, real-time capability can be achieved. The model equations are derived in Appendix 1.

Initial solution

The optimization problem requires an accurate initial solution $z_{b,\text{init}}$ to avoid convergence to an undesired local minimum. Similar to the works of Nishiwaki,[3,10] we make use of the so-called maximum kinematically feasible torso height $z_{\max}$ to derive the initial torso height trajectory (see Figure 6). The value of $z_{\max}$ is obtained using the kinematic chain from feet to torso while maintaining x_b and y_b. The vertical torso position can be increased depending on the position of the feet, x_{swing} and x_{stance}. The maximum kinematically feasible torso height per leg is reached for a fully stretched leg of length $L_{\max}$ with $\boldsymbol{q}_1 = 0$ according to equation (2). The overall maximum height $z_{\max}$ is the height that is feasible for both kinematic chains (equation (3)).

$$z_{fi,\max} = z_{fi} + \sqrt{L_{\max}^2 - (x_b - x_{fi})^2} \quad (2)$$

$$z_{\max} = \min_i (z_{fi,\max}) \quad (3)$$

The minimum feasible torso height $z_{\min}$ is obtained correspondingly with the ankle joint limit restricting the lowest possible configuration. These approximations are conservative since the 2-D model omits DoFs in the lateral direction and does not include hip and pelvis movements. These additional DoFs allow an even higher or lower torso height in 3-D. For calculation of an appropriate initial solution, timing of the control points and constraints have to be defined as follows.

Timing of set points. In the course of simulations, we found that the points in time of the control points are crucial for finding a feasible and optimal torso height trajectory. The torso height trajectory consists of four control points per step. The first and the last control points are set to the beginning and end of each step (see Figure 6). Due to limited calculation time, we also predefine the timing of two remaining control points. We identified two characteristic points in time for these control points inherent to each step that achieved satisfying results for our purposes: at the time of liftoff of the swing foot for upstairs or touchdown during downstairs movement (see Figure 6 for time t_1 and t_4), the maximum height $z_{\max}$ for a step reaches a local minimum.

The second intermediate control point considers the human ideal. In human walking, the torso is at the highest point during the single stance phase. This maximum occurs when the CoM is approximately above the swing foot[5] (see Figure 6 for time t_2 and t_5). Figure 6 shows the resultant choice of control points for two steps of moving up a platform.

Constraints for initial solution. The trajectory with N control points s_i is composed of $(N-1)$ cubic spline segments and has $4(N-1)$ parameters. There are necessary constraints to obtain a smooth trajectory: the first control point fulfills the boundary conditions to provide smooth C^2 connections from the previous step. To further enforce C^2 continuity, we introduce $3(N-2)$ continuity constraints on acceleration, velocity, and position level at the segment connections. For the initial solution $z_{b,\text{init}}$, we empirically define three additional constraints per step in order to avoid unfavorable local minima in the optimization. Figure 7 shows the torso height trajectory for stepping up a platform. The gray areas mark torso heights that violate joint limits according to the simplified model. An increasing platform height Δz_{stair} leads to a smaller range of feasible solutions.

We impose the constraints sequentially one step after another:

(1) To avoid the joint limits, a sufficient slope of the torso height trajectory is necessary. The minimum in $z_{\max}$ marks the point in time where this occurs (compare Figure 6: t_1 and t_4). The range of feasible solutions becomes narrow (compare Figure 7 at $t = 11.5$ s). Correspondingly, we determine the slope at these intermediate control points to be proportional to the difference in maximum torso height per step.

$$\dot{z}_{b,\text{init}}(t_1) = \frac{z_{\max}(t_3) - z_{\max}(t_0)}{\Delta t_{\text{step}}} p_{\text{tune}} \quad (4)$$

The tuning parameter p_{tune} is determined empirically and differs for upward and downward movements.

(2) Two more constraints are imposed onto the control point at the end of each step (i.e. at t_3 and t_6). Both, a change in step length and step height, affects the range of kinematically feasible torso height trajectories. We limit the height of the control points at t_3 and t_6 (see Figure 6). They represent the end points of a step movement. In the case of upstairs or downstairs movement, the control points at t_3 and t_6 are set higher or lower than the corresponding start point at t_0 with respect to t_3. An increase of the step length leads to a lower minimum in $z_{\max}$, since the feet are further apart. Accordingly, the height of the end points at t_3 and t_6 is adapted relative to the corresponding start point and the change in $z_{\max}$.

$$z_{b,\text{init}}(t_3) = z_0 + z_{\max}(t_3) - z_{\max}(t_0) \quad (5)$$

(3) We constrain the slope at the end points to achieve periodicity in the initial solution. In case neither step length nor height changed from the previous step, the slope at the end is constrained to the slope at the start of the step; otherwise, the slope is set to zero since no further insight about the slope is available.

$$\dot{z}_{b,\text{init}}(t_3) = \begin{cases} \dot{z}_0 & \text{same step parameters as previous step} \\ 0 & \text{else} \end{cases} \quad (6)$$

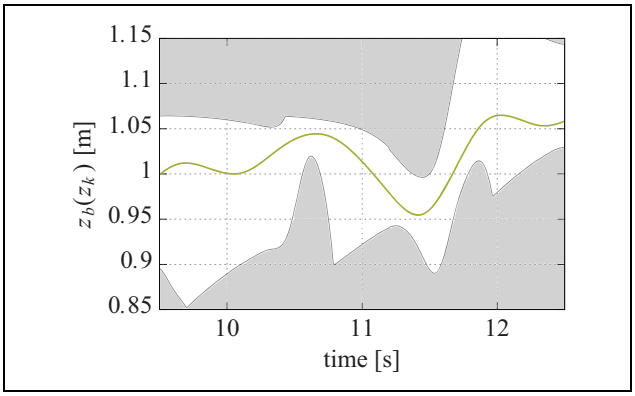

Figure 7. Range of feasible solutions $\Delta z_{\text{stair}} = 10$ cm. Gray areas mark torso heights greater than $z_{\max}$ and smaller than $z_{\min}$

The resultant cubic spline is uniquely defined with the complete set of $4(N-1)$ constraints.

Optimization

The previously defined initial trajectory serves as the starting point for a parameter optimization. Constraints imposed on the initial solution are omitted for the optimization. Only, the $3(N-2)$ conditions to ensure C^2-continuity as well as the three constraints providing smooth initial conditions are required. Consequently, there remain $N-1$ optimization parameters z_k per step period. This parameter set z_k is the minimal representation of the overall torso height trajectory z_b. The optimization of the parameters $z = [z_k]$ for k steps yields the local minimum with respect to a cost function $f(z)$. The optimization problem is defined as follows:

Cost function. The cost function $f(\mathbf{z})$ is a scalar function of the optimization parameters $\mathbf{z}$ with

$$f(z) = \int_0^{t_e} \left(w_{\dot{\boldsymbol{q}}}\, \dot{\boldsymbol{q}}(z)^T \dot{\boldsymbol{q}}(z) + w_{jl} H_{jl}(z) + w_{zm} H_{zm}(z) \right) \mathrm{d}t \tag{7}$$

The first term takes into account the joint velocities derived from the simplified 2-D model. H_{jl} and H_{zm} penalize the violation of joint limits and the violation of the maximum torso height with the weighting factors w_i.

The kinematic constraints are not included in the optimization problem as hard constraints due to real-time requirements. Since the 2-D model represents a conservative estimate of the real robot, violations of the kinematic constraints of the optimization model are acceptable for the full robot. Nevertheless, the optimization requires a well-chosen initial solution, which we are able to provide as explained in the previous section. The overall cost is obtained by numeric integration of the 2-D geometric model over a time horizon t_e.

Time horizon. The time horizon of the optimization problem is coupled to the robot's physical steps. By taking into account more than one step, future kinematic limits can be avoided. In this work, $n_{\text{steps}} \geq 1$ steps can be included in the trajectory optimization. Every additional step leads to an increased dimension of the optimization problem. As a trade-off between computational cost and prediction quality, we include $n_{\text{steps}} = 2$ steps in the optimization of the torso height trajectory.

Algorithm and real-time constraints. The optimization problem is solved using the open-source library for nonlinear optimization Nlopt (http://ab-initio.mit.edu/nlopt). The software package comes with a number of both global and local optimization routines. Experiments with different available algorithms showed that the sequential least squares programming (SLSQP) algorithm[25] and a sequential quadratic programming (SQP) gradient-based optimization algorithm, yields the best results for our purposes.

The torso height trajectory generation must be considerably shorter than one walking step. In the experiments, we limited the run-time to $T_{\text{opt}} = 400\,\text{ms}$ and keep the current best solution for the parameter set $\mathbf{z}$. Furthermore, we accelerate the model integration by increasing the integration time step to $\Delta t_{\text{exp}} = 8\Delta t$ with $\Delta t = 1.5\,\text{ms}$. Its influence of the solution quality is analyzed in the following section.

Results

We have assessed the performance of the newly proposed torso height trajectory generation using the dynamics simulation framework described in the work by Buschmann et al.[2] and validated our approach in experiments (a video of our experiments is available: https://youtu.be/ayj95PVvq0 M). Our approach shows improved performance in challenging stepping scenarios involving obstacles, platforms, or stairs compared with our previous results. Even walking, especially fast walking, also yielded cost reductions for the full robot motions. The presented simulation results compare the newly proposed method to generate $z_b(z_k)$ with optimized control points z_k ("on") and the torso height trajectory $z_b(H)$, where the final torso height position H is optimized as part of the *Parameter Optimization* ("off").

Platform

In this scenario, a platform of height $\Delta z_{\text{stair}} = 12.5\,\text{cm}$ is placed in front of the robot. Figure 8 depicts *Lola* in front of the platform in simulation and in experiment. *Lola* stepping up and down the platform in experiments is shown in Figure 9.

Validation of kinematic model. The performance of the proposed trajectory optimization strongly depends on the validity of the presented 2-D kinematic model. The ankle joint angles of the simple 2-D model follow the full kinematics well. Comparing the knee joint angles in Figure 10 shows that the 2-D model differs by more than 15° compared to the 3-D model when the robot steps onto the platform (compare time approximately 10.5 s). During this movement, the DoFs in the hip and pelvis, which are not represented in the reduced model, play an important role. Regarding the knee joint angle, the 2-D model approximation is conservative toward the lower joint limit. The simplified model angle could run into the limits while the real joint angle is still in the working range. To exploit this open optimization potential, we reduce the weight factor w_{jl} for the knee joint limits in the optimization.

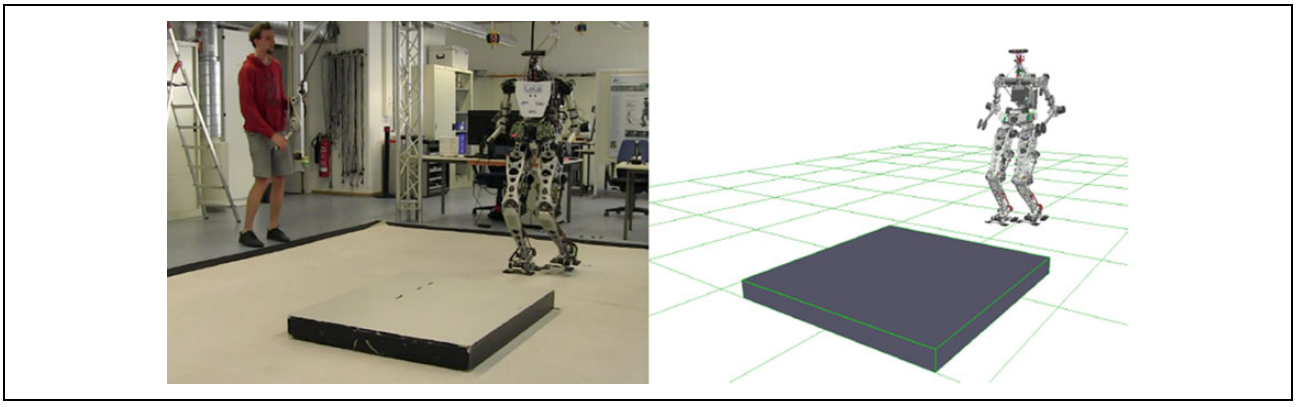

Figure 8. Platform test case: snapshots showing *Lola* in experiment (left) and in simulation (right) walking ahead to platform.

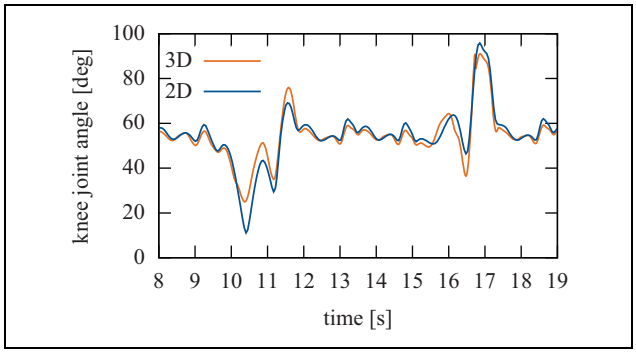

Figure 9. Validation of kinematic model: left knee joint model comparison.

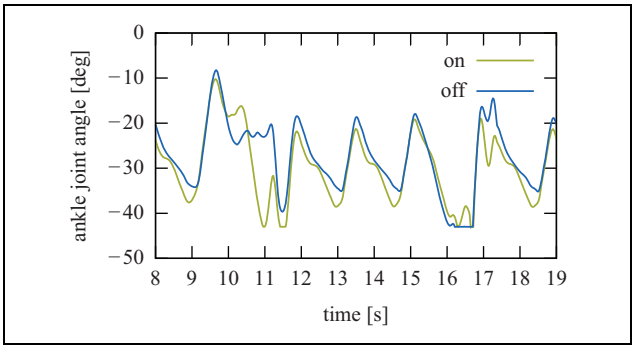

Figure 11. Platform test case: left ankle angle with optimization of final torso height position ("off") and with proposed method ("on")

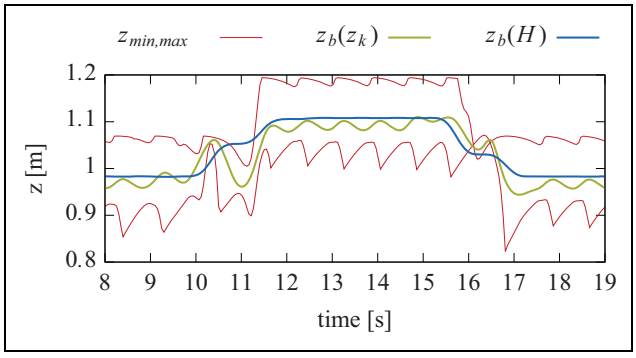

Figure 10. Platform test case: torso height trajectories.

Results of simulation and experiment. Figure 11 shows the resulting torso height trajectories using the proposed method "on" and "off." Furthermore, the upper and lower limits of the torso height trajectory calculated with the 2-D model are shown. It is clearly visible, how the torso trajectory with optimization of the support points outperforms the former torso height trajectory by abiding the kinematic constraints. When stepping down the platform, the ankle joint limit of the back leg is reached when only the final torso position H is optimized (Figure 12). The mechanical limit is avoided by limiting the executable joint angle resulting in the constant joint angle between $t = 16\,\mathrm{s}$ and $t = 17\,\mathrm{s}$ in Figure 12. This hard restriction affects the smoothness of moving up and, especially, down platforms. The optimized trajectory $z_b(z_k)$ avoids the ankle joint limit when stepping down. This is a major improvement over the former implementation and results in a much smoother stepping down movement. When stepping down, there remains no valid range for the torso height to stay within joint limits of the 2-D kinematic model ($t = 16\,\mathrm{s}$ in Figure 11). As we learned by validating the 2-D model, it approximates the knee joint value conservatively. With the reduced weight factor w_{jl} for the knee joint limit, the optimization yields a viable solution for the 3-D case. In both approaches to derive the torso height trajectory, the knee joint limits are not violated.

Podium. This test case consists of a podium with two consecutive stairs of height $\Delta z_{\mathrm{stair}} = 10\,\mathrm{cm}$ up and down. Figure 13 depicts *Lola* stepping dynamically up and down the podium. We have not validated this test case in experiments. Nevertheless, according to the quality of our simulation and experimental proof for the platform test case, we are confident that this scenario is manageable in the experiment as well. The resulting torso height trajectories are

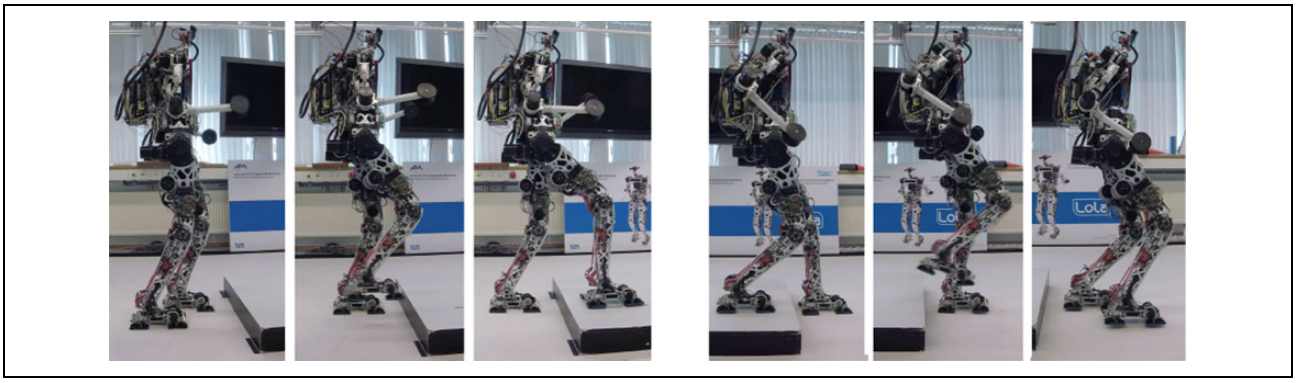

Figure 12. Experiment-stepping up and down: snapshots showing Lola stepping dynamically up and down a platform using the proposed method.

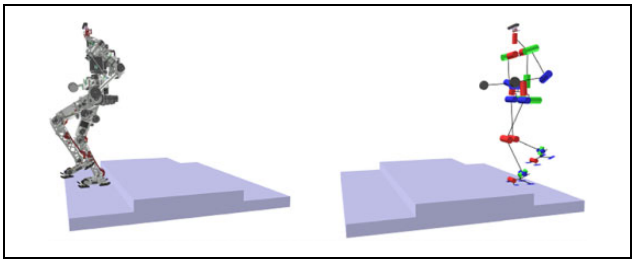

Figure 13. Podium test case: snapshots showing *Lola* stepping dynamically up and down a podium with two different heights. Left: *Lola* fully modeled. Right: *Lola*'s kinematic model.

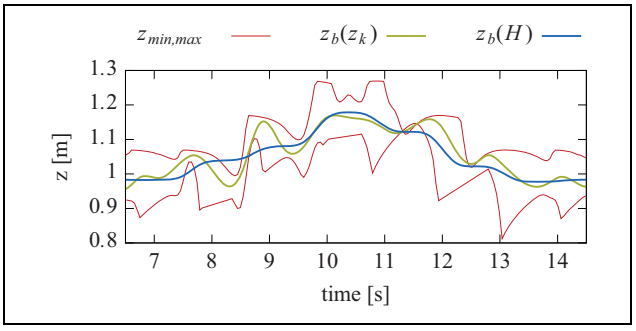

Figure 14. Podium test case: torso height trajectories.

shown in Figure 14. Similar to the platform test case, it is clearly visible that the torso height trajectory calculated using the proposed method performs best. Furthermore, we compared the torso trajectories calculated using the 2-D model with Δt and $\Delta t_{exp} = 8\Delta t$. Figure 15 shows that this decrease of accuracy leads to only slightly different trajectory in this complex stepping scenario.

Fast walking

The third test case is a simple walking scenario. It consists of a sequence of steps with a step size of up to $l_x = 0.5\,\mathrm{m}$ and a step time of $T_S = 0.7\,\mathrm{s}$. Figure 16 shows the torso height trajectory with optimized end points in comparison to the proposed cubic spline obtained via optimization. The

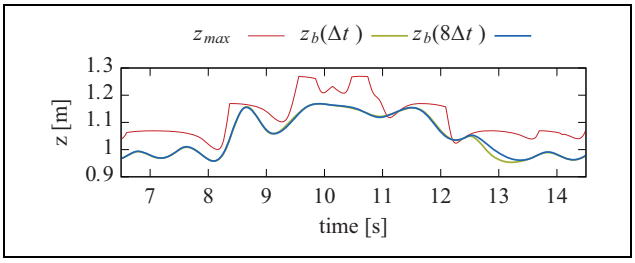

Figure 15. Comparison of integration step size of 2-D kinematic model.

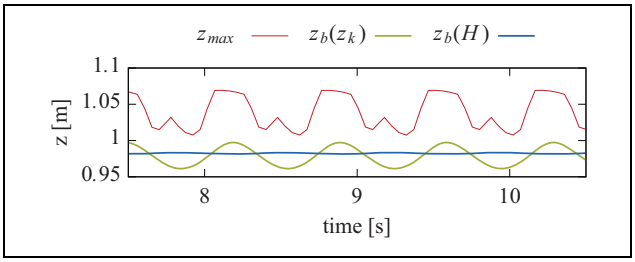

Figure 16. Fast walking test case: torso height trajectories.

resultant optimized trajectory strongly resembles the cosine torso height that can be observed in human walking. The corresponding joint velocity costs $w\dot{\boldsymbol{q}}\dot{\boldsymbol{q}}^T\dot{\boldsymbol{q}}$ are depicted in Figure 17. A cost reduction is clearly visible. This confirms that the optimization performed using the reduced 2-D model also yields reduced velocities considering the full kinematics and all robot joints.

Contribution and outlook

In this article, we focus on methods to improve the kinematic versatility of bipedal walking. We modify the collocation method presented by Buschmann et al.[22] to introduce a more generally shaped trajectory for the torso height for bipedal locomotion. The torso height is taken into account explicitly in the CoM trajectory generation and, therefore, does not diminish the robot's stability. A

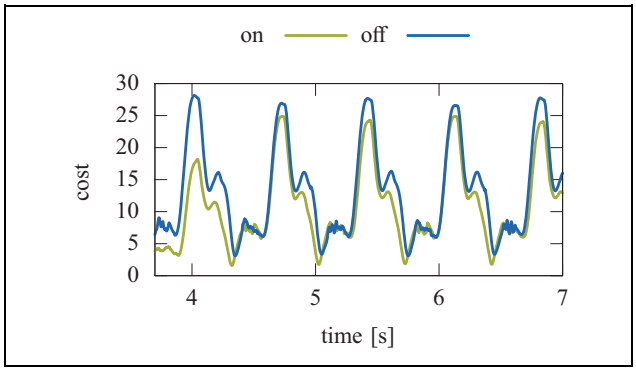

Figure 17. Experiment—fast walking test case: joint velocity costs with optimization of final torso height position ("off") and with proposed method ("on").

new parametrization for the torso height based on cubic spline representation is applied. For real-time application, the robot's full kinematic model is approximated by a simplified 2-D model that allows us for a larger number of optimization parameters and a longer time horizon of multiple walking steps. The model is verified in simulations for different walking scenarios. In combination with our methods for collision avoidance, it also allows for collision-free motions. It is verified in simulations and validated in experiments. Using the presented strategy, the robot is able to walk in complex scenarios (e.g. platforms, obstacles) with reduced joint velocities and safer joint movement ranges (with respect to joint limits). In future work, we want to analyze the presented method in more complex scenarios. We aim to overcome current real-time constraints, to combine the presented vertical trajectory optimization with the *Parameter Optimization* to allow for a more holistic pattern generation. Furthermore, the point in time of the control points is a critical factor in the performance of the method. Our aim is to include it in the optimization.

Acknowledgements

This work was supported by the German Research Foundation (DFG) and the Technical University of Munich (TUM) in the framework of the Open Access Publishing Program.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of following financial support for the research, authorship, and/or publication of this article: This work is supported by the German Academic Exchange Service (DAAD), the German Research Foundation (DFG)—project BU 2736/1-1- and the Technical University of Munich (TUM) in the framework of the Open Access Publishing Program.

ORCID iD

Arne-Christoph Hildebrandt http://orcid.org/0000-0002-9127-8344

References

1. Takenaka T, Matsumoto T, and Yoshiike T. 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*, St. Louis, MO, USA, 10–15 October 2009.
2. Buschmann T, Favot V, Lohmeier S, et al. Experiments in fast biped walking. In: *IEEE international conference on mechatronics*, Istanbul, Turkey, 13–15 April 2011.
3. Nishiwaki K, Chestnutt J, and Kagami S. Planning and control of a humanoid robot for navigation on uneven multi-scale terrain. In: *Springer tracts in advanced robotics*, vol. 79, Berlin, 2014, pp. 401–415.
4. Herdt A, Perrin N, and Wieber PB. LMPC based online generation of more efficient walking motions. In: *IEEE-RAS international conference on humanoid robots*, Osaka, Japan, 29 November–1 December 2012.
5. Li Z, Vanderborght B, Tsagarakis NG, et al. Fast bipedal walk using large strides by modulating hip posture and toe heel motion. In: *IEEE international conference on robotics and biomimetics*, Tianjin, China, 14–18 December 2010.
6. Chevallereau C and Aoustin Y. Self-stabilization of 3D walking via vertical oscillations of the hip. In: *IEEE international conference on robotics and automation*, Seattle, WA, USA, 26–30 May 2015.
7. Koolen T, Posa M, and Tedrake R. Balance control using center of mass height variation: limitations imposed by unilateral contact. In: *IEEE-RAS international conference on humanoid robots*, Cancun, Mexico, 15–17 November 2016.
8. Hildebrandt AC, Wahrmann D, Wittmann R, et al. Realtime pattern generation among obstacles for biped robots. In: *IEEE/RSJ international conference on intelligent robots and systems*, Hamburg, Germany, 28 September–2 October 2015.
9. Kajita S, Kanehiro F, Kaneko K, et al. Biped walking pattern generation by using preview control of zero-moment point. In: *IEEE international conference on robotics and automation*, Taipei, Taiwan, 14–19 September 2003.
10. Nishiwaki K. Online design of torso height trajectories for walking patterns that takes future kinematic limits into consideration. In: *IEEE international conference on robotics and automation*, Shanghai, China, 9–13 May 2011.
11. Mayr J, Gattringer H, and Bremer H. A bipedal walking pattern generator that considers multi-body dynamics by angular momentum estimation. In: *IEEE-RAS international conference on humanoid robots*, Osaka, Japan, 29 November–1 December 2012.
12. Brasseur C, Sherikov A, Collette C, et al. A robust linear MPC approach to online generation of 3D biped walking motion. In: *IEEE-RAS international conference on humanoid robots*, Seoul, South Korea, 3–5 November 2015.
13. Shafii N, Lau N, and Reis LP. Learning a fast walk based on ZMP control and hip height movement. In: *IEEE international conference on autonomous robot systems and competitions*, Espinho, Portugal, 14–15 May 2014.

14. Park CS, Ha T, Kim J, et al. Trajectory generation and control for a biped robot walking upstairs. *Int J Control Autom Syst* 2010; 8(2): 339–351.
15. Hong YD and Lee KB. Stable walking of humanoid robots using vertical center of mass and foot motions by an evolutionary optimized central pattern generator. *Int J Adv Robot Syst* 2016; 13(1): 27.
16. Miura K, Morisawa M, Kanehiro F, et al. Human-like walking with toe supporting for humanoids. In: *IEEE international conference on intelligent robots and systems*, San Francisco, CA, USA, 25–30 September 2011.
17. Griffin RJ, Wiedebach G, Bertrand S, et al. Straight-leg walking through underconstrained whole-body control. In: *IEEE international conference on robotics and automation (ICRA)*, Brisbane, Australia, 21–25 May 2018.
18. Hu K, Ott C, and Lee D. Online human walking imitation in task and joint space based on quadratic programming. In: *IEEE international conference on robotics and automation*, Brisbane, Australia, 21–25 May 2018.
19. Kuindersma S, Deits R, Fallon M, et al. Optimization-based locomotion planning, estimation, and control design for Atlas. *Auton Robot* 2016; 40(3): 429–455.
20. Carpentier J, Tonneau S, Naveau M, et al. A versatile and efficient pattern generator for generalized legged locomotion. In: *IEEE international conference on robotics and automation*, Stockholm, Sweden, 16–21 May 2016.
21. Hildebrandt AC, Demmeler M, Wittmann R, et al. Real-time predictive kinematic evaluation and optimization for biped robots. In: *IEEE/RSJ international conference on intelligent robots and systems*, Daejeon, South Korea, 9–14 October 2016.
22. Buschmann T, Lohmeier S, Bachmayer M, et al. A collocation method for real-time walking pattern generation. In: *IEEE/RAS international conference on humanoid robots*, Pittsburgh, PA, USA, 29 November–1 December 2007.
23. Wahrmann D, Hildebrandt AC, Wittmann R, et al. Fast object approximation for real-time 3D obstacle avoidance with biped robots. In: *IEEE international conference on advanced intelligent mechatronics*, Banff, AB, Canada, 12–15 July 2016.
24. Wittmann R, Hildebrandt AC, Wahrmann D, et al. Real-time nonlinear model predictive footstep optimization for biped robots. In: *IEEE-RAS international conference on humanoid robots*, Seoul, South Korea, 3–5 November 2015.
25. Kraft D.Algorithm 733; TOMP–Fortran modules for optimal control calculations. *ACM Trans Math Softw* 1994; 20(3): 262–281.

Appendix 1

The 2-D kinematic model is best understood as a triangle with hip, knee, and ankle in the corners. The lengths of two of the edges are fixed by the robot's dimensions. The third edge is the distance h between hip and ankle. The projected distances h_x and h_z can be obtained from robot dimensions and the trajectories for feet and body. The law of cosines yields the angles α, γ_1, and γ_2 in the triangle between hip, knee, and ankle joint equations (1A) to (1C).

$$\alpha = \operatorname{acos}\left(\frac{l_1^2 + l_2^2 - h^2}{2l_1 l_2}\right) \tag{1A}$$

$$\gamma_1 = \operatorname{acos}\left(\frac{h^2 + l_2^2 - l_1^2}{2hl_2}\right) \tag{1B}$$

$$\gamma_2 = \operatorname{acos}\left(\frac{l_1^2 + h^2 - l_2^2}{2l_1 h}\right) \tag{1C}$$

Together with the sign of h_x, the distance between ankle joint and hip joint projected in the forward pointing x direction, the three leg joint angles can be derived from the angles in equations (1A) to (1C) and some basic trigonometry (equation (1D)).

$$\begin{aligned} h_x > 0 \quad q_0 &= -\gamma_2 - \operatorname{acos}\left(\frac{h_z}{h}\right) \\ q_1 &= 180^\circ - \alpha \\ q_2 &= 180^\circ - \gamma_1 + \operatorname{atan}\left(\left\|\frac{h_x}{h_z}\right\|\right) \end{aligned} \tag{1D}$$