

# Torso Height Optimization for Bipedal Locomotion

Arne-Christoph Hildebrandt, Konstantin Ritt, Daniel Wahrmann, Robert Wittmann,  
Felix Sygulla, Philipp Seiwald, Daniel Rixen, and Thomas Buschmann<sup>1</sup>

**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 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 in order 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.

## I. 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 (CoM) resp. 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 paper, 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 reaching kinematic constraints. In the next section, we present related work and highlight the originality of our contribution. The following section provides an overview of the experimental platform used in this work – the robot *Lola* (Fig. 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 as well as the optimization is evaluated

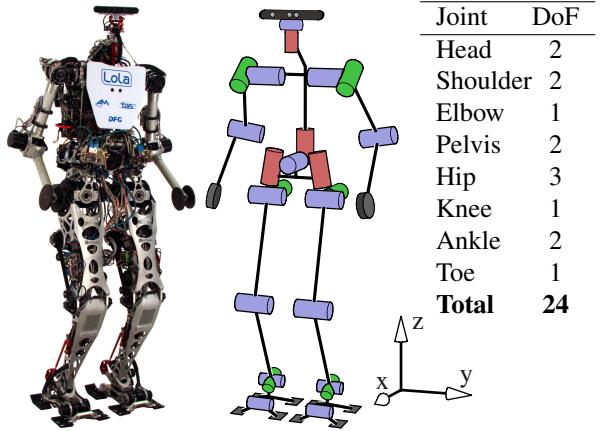


Fig. 1: Photo and kinematic structure of the humanoid robot *Lola*. The joint distribution and world coordinate system used are shown on the right side (adapted from [8]).

in simulations and validated in successfully conducted experiments. The paper concludes with a conclusion and comments on future work.

## II. 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 multi-body system. Usually, these trajectories include the center of mass 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 ([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 by [1], [10], [11], among others. 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 [5], [6], [12], 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. [13] use a 6th order polynomial to generate the CoM height trajectory. They define boundary conditions and design parameters that are

<sup>1</sup>Chair of Applied Mechanics, Technical University of Munich, 85747 Garching, GER, Email: arne.hildebrandt@tum.de

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.

The authors of [14] 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. [15] 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.

The authors of [10] propose a similar approach. 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, taking 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 [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 [16], [17]. These approaches are either pure offline methods [17] or still need initially long calculation times [16].

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 [18]. Using the strategy presented in [19], the torso height is explicitly taken into account to generate dynamically feasible CoM motions.

In contrast to [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,

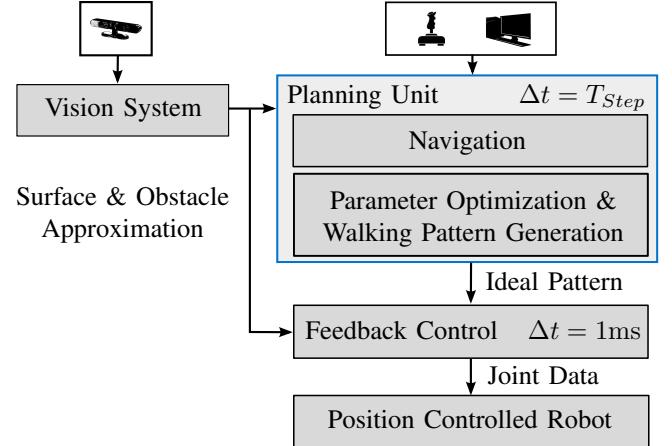


Fig. 2: *Lola*'s real-time walking control system.

taking into account kinematic limitations and joint velocities. The model is verified and the optimization is analyzed in simulations and validated in experiments.

### III. CONTROL ARCHITECTURE

We use our humanoid robot *Lola* for the method's validation. Fig. 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 in [2].

#### A. Control Architecture

An overview of our hierarchical walking control system is depicted in Fig. 2. The *Vision System* [20] uses only the on-board camera and approximates the environment with swept sphere volumes (SSV). The representation of the environment and of the robot via SSV objects is used consistently in all control modules for fast distance calculations. The robot control is divided into a high-level *Planning Unit* which is executed once every walking step  $k$  (step time of  $T_{Step} = 0.6\text{--}1.2\text{s}$ ), and a *Feedback Control* with a cycle time of 1ms. The *Planning Unit* first calculates, in the *Navigation* module [8], a sequence of parameter sets  $p_{wp,k}$  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  $p_{wp,k}$  the *Parameter Optimization & Walking Pattern Generation* evaluates and optimizes  $p_{wp,1}$  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.

#### B. Center of Mass Trajectory Generation

The horizontal CoM trajectories are generated as part of the walking pattern generation using a three-mass model

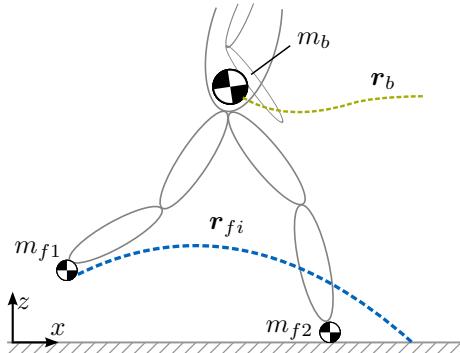


Fig. 3: Three-mass model used for CoM trajectory calculation [21].

to account for dynamic effects caused by fast leg movements (see Fig. 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  $r_{fi}(\mathbf{p}_{wp}, t) = [x_{fi}(\mathbf{p}_{wp}, t), y_{fi}(\mathbf{p}_{wp}, t), z_{fi}(\mathbf{p}_{wp}, t)]$ , the desired torso height trajectory  $z_b = z_b(\mathbf{p}_{wp}, t)$  describing the movement of the upper body mass  $m_b$ , and the desired contact moments trajectories  $T_x$  and  $T_y$ . The torso height  $z_b = z_b(\mathbf{p}_{wp}, t) = z_b(H, t)$  is a fifth order polynomial with a configurable height  $H$  at the end of each step.<sup>1</sup>

Thus, the equation of motion 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

$$\begin{aligned} m_b z_b \ddot{y}_b - m_b y_b (\ddot{z}_b + g) &= -T_x \\ + m_f y_{fi} (\ddot{z}_f + g) - m_f z_{fi} \ddot{y}_{fi} & \quad (1) \\ + m_f y_{f2} (\ddot{z}_f + g) - m_f z_{f2} \ddot{y}_{f2} \end{aligned}$$

The equation for the sagittal plane can be derived analogously. We use the method based on spline collocation proposed in [19] 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 [19]. The overall CoM trajectories  $\mathbf{r}_{CoM}$  can be calculated by superposition of the motions of the masses  $m_b$  and  $m_{fi}$ .

### C. Kinematic Evaluation & Parameter Optimization

The parameter set  $\mathbf{p}_{wp}$  configures the walking pattern of the robot and governs the robot's motion. Offline defined parameters are not optimal with respect to dynamic and complex scenarios. Parameters which 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 [18], we introduced the *Parameter Optimization* (see Fig. 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,  $\mathbf{p}_{wp}$  is evaluated and optimized in real-time.

1) *Limitations of 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<sup>2</sup>.

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.

## IV. 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.

Fig. 4 gives an overview of how the proposed method extends the current pattern generation procedure. In Sub-section IV-A, we present a new torso height trajectory parametrization which allows for a more sophisticated trajectory design and which can be configured with more than one parameter per step. 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 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 [18]. This is a trade-off between optimality of the solution and calculation time for real-time application.

### A. Parametrization 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 Fig. 6). This representation provides  $C^2$ -smooth

<sup>1</sup>For the sake of simplicity, we will omit the explicit dependency on  $\mathbf{p}_{wp}$  in the following.

<sup>2</sup>A video showing *Lola* stepping up and down a platform with the current trajectory design is available at <https://youtu.be/rKsx8HKvBkg>.

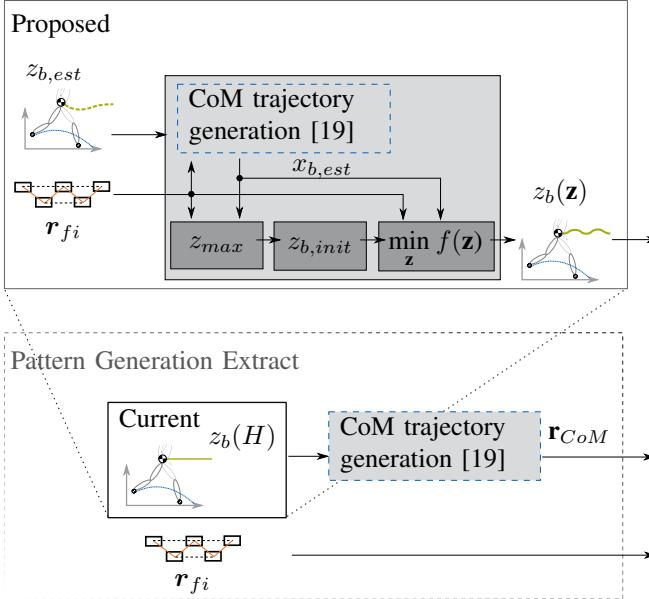


Fig. 4: Overview of proposed method in the context of the pattern generation process. This figure focuses on the optimization process of the torso height.

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 one 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  $z_k, k = 0..3$ . 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.

### B. 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 2D model in the sagittal plane. Thus, the 2D kinematic chain of each leg is fully determined with the trajectories  $\mathbf{r}_b = [x_b, y_b, z_b]$ ,  $\mathbf{r}_{fi}$ ,  $\varphi$ , and the heuristically defined toe trajectory  $\varphi_{toe}$ . Fig. 5 shows a 2D 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  $\mathbf{r}_{fi}$ ,  $\varphi$ , and  $\varphi_{toe}$  are defined by  $p_{wp}$ . The horizontal trajectory of the body mass point of the three-mass model  $x_b$  and  $y_b$  result from solving the governing Equations of Motion (EoMs) (see (1)). As explained

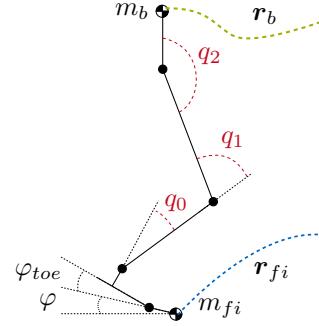


Fig. 5: 2D kinematic model approximating robot's full kinematics. Input trajectories:  $\mathbf{r}_b$ ,  $\mathbf{r}_{fi}$ ,  $\varphi$ , and  $\varphi_{toe}$ . Unknown: joint angles in red.

in the previous section, the torso height trajectory  $z_b$  itself is a necessary input to solve these equations. An estimated torso height trajectory  $z_{b,est}$  is used to solve the EoMs to yield a sufficiently accurate horizontal trajectory  $x_{b,est}$ . When  $\mathbf{r}_b$ ,  $\mathbf{r}_{fi}$ ,  $\varphi$ , and  $\varphi_{toe}$  are determined, the inverse kinematics can be solved analytically for the joint angles  $q$ . The joint velocities  $\dot{q}$  are derived numerically. Thus, real-time capability can be achieved. The model equations are derived in the Appendix.

### C. Initial Solution

The optimization problem requires an accurate initial solution  $z_{b,init}$  to avoid convergence to an undesired local minimum. Similar to [3], [10], we make use of the so called maximum kinematically feasible torso height  $z_{max}$  to derive the initial torso height trajectory (see Fig. 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  $q_1 = 0$  according to equation (2). The overall maximum height  $z_{max}$  is the height that is feasible for both kinematic chains (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 2D 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 3D. For calculation of an appropriate initial solution, timing of the control points and constraints have to be defined as follows.

*1) 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 Fig. 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 lift-off of the swing foot for upstairs or touchdown during downstairs movement (see Fig. 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 Fig. 6 for time  $t_2$  and  $t_5$ ). Fig. 6 shows the resultant choice of control points for two steps of moving up a platform.

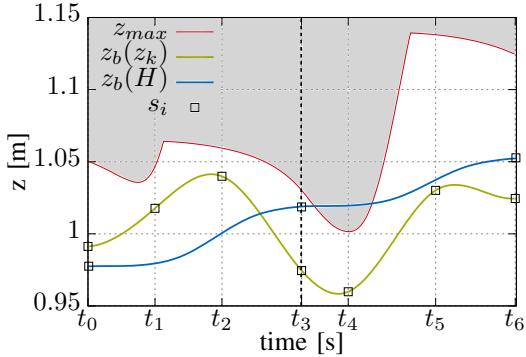


Fig. 6: Walking up a platform with  $\Delta z_{stair} = 7.5\text{cm}$ : control points of torso height trajectories for two walking steps (separated by dotted line). Blue line denotes trajectory resulting from optimization of  $H$ , green line denotes trajectory resulting from presented method.

2) *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  $z_{b,init}$  we empirically define three additional constraints per step in order to avoid unfavorable local minima in the optimization. Fig. 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  $\Delta z_{stair}$  leads to a smaller range of feasible solutions.

We impose the constraints sequentially one step after another:

(1) In order 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 Fig. 6:  $t_1$  and  $t_4$ ). The range of feasible solutions becomes narrow (compare Fig. 7 at  $t=11.5\text{s}$ ). Correspondingly, we determine the slope at these intermediate control points to be proportional to the difference in maximum torso height per step.

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

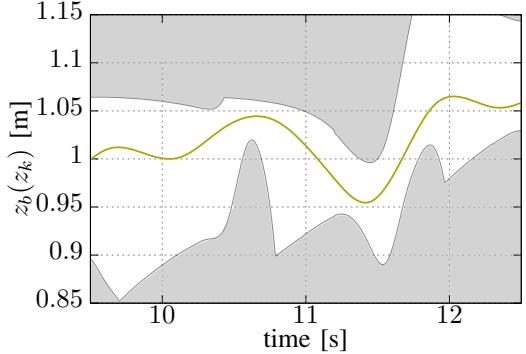


Fig. 7: Range of feasible solutions  $\Delta z_{stair} = 10\text{cm}$ . Gray areas mark torso heights greater than  $z_{max}$  and smaller than  $z_{min}$  that violate joint limits.

The tuning parameter  $p_{tune}$  is determined empirically and differs for up and downwards 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, affect the range of kinematically feasible torso height trajectories. We limit the height of the control points at  $t_3$  and  $t_6$  (see Fig. 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  $t_0$  resp.  $t_3$ . 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  $z_{max}$ .

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

(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.

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

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

#### D. 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:

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

$$f(\mathbf{z}) = \int_0^{t_e} \left( w_{\dot{q}} \dot{q}(\mathbf{z})^T \dot{q}(\mathbf{z}) + w_{jl} H_{jl}(\mathbf{z}) + w_{zm} H_{zm}(\mathbf{z}) \right) dt. \quad (7)$$

The first term takes into account the joint velocities derived from the simplified 2D 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 2D 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 2D geometric model over a time horizon  $t_e$ .

2) *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,  $n_{steps} \geq 1$  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  $n_{steps} = 2$  steps in the optimization of the torso height trajectory.

3) *Algorithm and Real-Time Constraints*: The optimization problem is solved using the open-source library for nonlinear optimization NLOpt<sup>3</sup>. The software package comes with a number of both global and local optimization routines. Experiments with different available algorithms showed that the SLSQP algorithm [22], 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  $T_{opt} = 400\text{ms}$  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  $\Delta t_{exp} = 8\Delta t$  with  $\Delta t = 1.5\text{ms}$ . Its influence of the solution quality is analyzed in the following section.

## V. RESULTS

We have assessed the performance of the newly proposed torso height trajectory generation using the dynamics simulation framework described in [2] and validated our approach in experiments<sup>4</sup>. 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").

### A. Platform

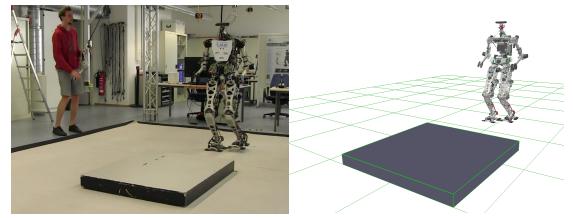


Fig. 8: Platform test case: snapshots showing *Lola* in experiment (left) and in simulation (right) walking ahead to platform.

In this scenario, a platform of height  $\Delta z_{stair} = 12.5\text{cm}$  is placed in front of the robot. Fig. 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 Fig. 12.

1) *Validation of Kinematic Model*: The performance of the proposed trajectory optimization strongly depends on the validity of the presented 2D kinematic model. The ankle joint angles of the simple 2D model follow the full kinematics well. Comparing the knee joint angles in Fig. 9 shows that

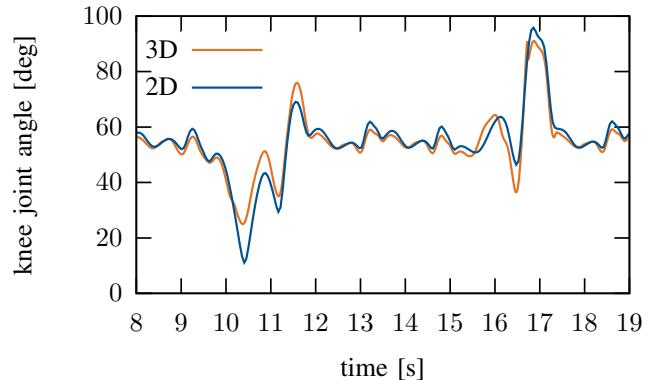


Fig. 9: Validation of kinematic model: left knee joint model comparison.

the 2D model differs by more than  $15^\circ$  compared to the 3D model when the robot steps onto the platform (compare time approx. 10.5s). 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 2D model approximation is conservative towards 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.

2) *Results of Simulation and Experiment*: Fig. 10 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 2D model are shown. It is clearly visible, how the torso trajectory with optimization of the support points outperforms the former

<sup>3</sup><http://ab-initio.mit.edu/nlopt>

<sup>4</sup>A video of our experiments is available: <https://youtu.be/ayj95PVvq0M>.

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 (Fig. 11). The mechanical limit is avoided by limiting the executable joint angle resulting in the constant joint angle between  $t = 16s$  and  $t = 17s$  in Fig. 11. This hard restriction affects the smoothness of moving up and especially down platforms. The optimized trajectory  $z_b(z_k)$

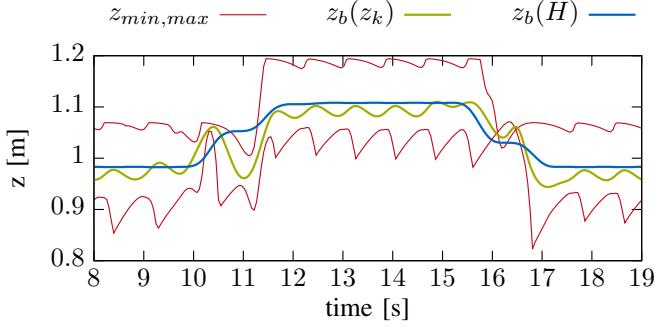


Fig. 10: Platform test case: torso height trajectories.

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 2D kinematic model ( $t = 16s$  in Fig. 10). As we learned from validating the 2D 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 3D case. In both approaches to derive the torso height trajectory the knee joint limits are not violated.

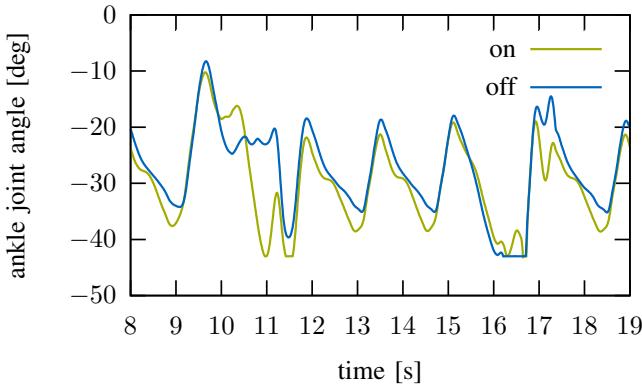


Fig. 11: Platform test case: left ankle angle with optimization of final torso height position ("off") and with proposed method ("on").

3) *Podium:* This test case consists of a podium with two consecutive stairs of height  $\Delta z_{stair} = 10cm$  up and down. Fig. 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 experiment as well.

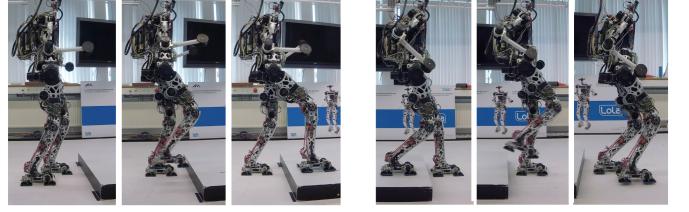


Fig. 12: Experiment – stepping up and down: snapshots showing *Lola* stepping dynamically up and down a platform using the proposed method.

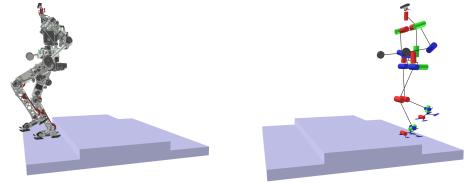


Fig. 13: Podium test case: snapshots showing *Lola* stepping dynamically up and down a podium with two different heights. Left: *Lola* fully modeled. Right: *Lola*'s kinematic model.

The resulting torso height trajectories are shown in Fig. 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

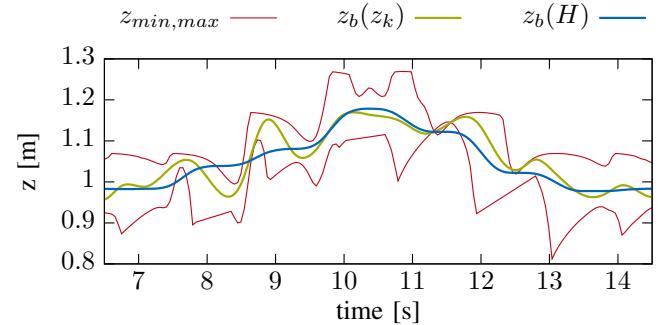


Fig. 14: Podium test case: torso height trajectories.

calculated using the 2D model with  $\Delta t$  and  $\Delta t_{exp} = 8\Delta t$ . Fig. 15 shows that this decrease of accuracy leads to an only slightly different trajectory in this complex stepping scenario.

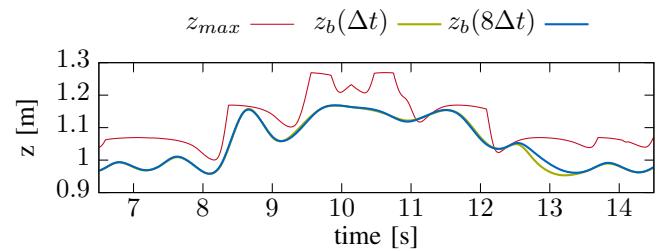


Fig. 15: Comparison of integration step size of 2D kinematic model.

## B. 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  $l_x = 0.5m$  and a step time of  $T_S = 0.7s$ . Fig. 16 shows the torso height trajectory with optimized end points in comparison to the proposed cubic spline obtained via optimization. The resultant optimized trajectory strongly resembles the cosine torso height that can be observed in human walking. The corresponding

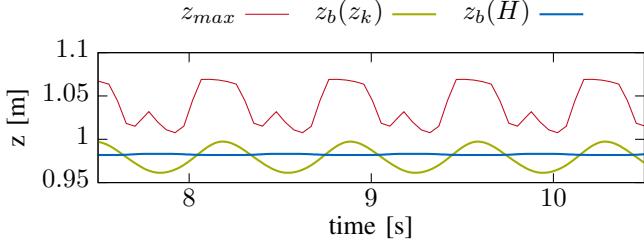


Fig. 16: Fast walking test case: torso height trajectories.

joint velocity costs  $w_q \dot{q}^T \dot{q}$  are depicted in Fig. 17. A cost reduction is clearly visible. This confirms that the optimization performed using the reduced 2D model also yields reduced velocities considering the full kinematics and all robot joints.

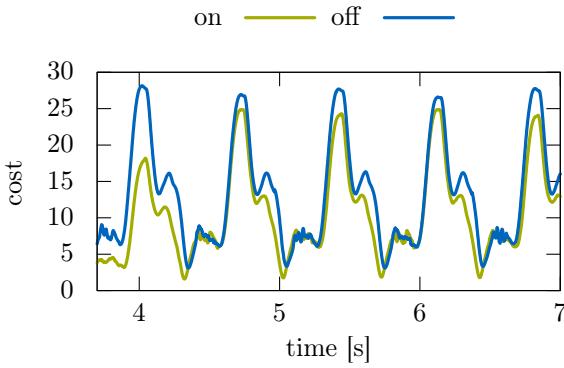


Fig. 17: Experiment – fast walking test case: joint velocity costs with optimization of final torso height position ("off") and with proposed method ("on").

## VI. CONTRIBUTION AND OUTLOOK

In this paper, we focus on methods to improve the kinematic versatility of bipedal walking. We modify the collocation method presented in [19] 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 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 2D 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. 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.

## VII. APPENDIX

The 2D 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  $\alpha$ ,  $\gamma_1$  and  $\gamma_2$  in the triangle between hip, knee and ankle joint (8)-(10).

$$\alpha = \arccos\left(\frac{l_1^2 + l_2^2 - h^2}{2l_1l_2}\right) \quad (8)$$

$$\gamma_1 = \arccos\left(\frac{h^2 + l_2^2 - l_1^2}{2hl_2}\right) \quad (9)$$

$$\gamma_2 = \arccos\left(\frac{l_1^2 + h^2 - l_2^2}{2l_1h}\right) \quad (10)$$

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 (8)-(10) and some basic trigonometry (11).

$$\begin{aligned} h_x > 0 & \quad q_0 = -\gamma_2 - \arccos\left(\frac{h_z}{h}\right) \\ q_1 &= 180^\circ - \alpha \\ q_2 &= 180^\circ - \gamma_1 + \arctan\left(\frac{h_x}{h_z}\right) \end{aligned} \quad (11)$$

## ACKNOWLEDGMENTS

This work is supported by the DAAD and the Deutsche Forschungsgemeinschaft (DFG project BU 2736/1-1).

## REFERENCES

- [1] 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.
- [2] T. Buschmann, V. Favot, S. Lohmeier, M. Schwienbacher, and H. Ulbrich, "Experiments in Fast Biped Walking," in *IEEE International Conference on Mechatronics*, 2011.

- [3] K. Nishiwaki, J. Chestnutt, and S. Kagami, "Planning and Control of a Humanoid Robot for Navigation on Uneven Multi-scale Terrain," in *Springer Tracts in Advanced Robotics*, vol. 79, Berlin, Heidelberg, 2014, pp. 401–415.
- [4] A. Herdt, N. Perrin, and P. B. Wieber, "LMPC based online generation of more efficient walking motions," in *IEEE-RAS International Conference on Humanoid Robots*, 2012.
- [5] Z. Li, B. Vanderborght, N. G. Tsagarakis, and D. G. Caldwell, "Fast bipedal walk using large strides by modulating hip posture and toe-heel motion," in *IEEE International Conference on Robotics and Biomimetics*, 2010.
- [6] C. Chevallereau and Y. Aoustin, "Self-Stabilization of 3D Walking via Vertical Oscillations of the Hip," in *IEEE International Conference on Robotics and Automation*, 2015.
- [7] T. Koolen, M. Posa, and R. Tedrake, "Balance control using center of mass height variation: limitations imposed by unilateral contact," in *IEEE-RAS International Conference on Humanoid Robots*, 2016.
- [8] A.-C. Hildebrandt, D. Wahrmann, R. Wittmann, D. Rixen, and T. Buschmann, "Real-Time Pattern Generation Among Obstacles for Biped Robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015. [Online]. Available: <https://youtu.be/W-jIxVahyI>.
- [9] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped Walking Pattern Generation by Using Preview Control of Zero-Moment Point," in *IEEE International Conference on Robotics and Automation*, 2003.
- [10] K. Nishiwaki, "Online design of torso height trajectories for walking patterns that takes future kinematic limits into consideration," in *International Conference on Control, Automation and Systems*, 2011.
- [11] J. Mayr, H. Gatringer, and H. Bremer, "A bipedal walking pattern generator that considers multi-body dynamics by angular momentum estimation," in *IEEE-RAS International Conference on Humanoid Robots*, 2012.
- [12] N. Shafii, N. Lau, and L. P. Reis, "Learning a fast walk based on ZMP control and hip height movement," in *IEEE International Conference on Autonomous Robot Systems and Competitions*, 2014.
- [13] C.-S. Park, T. Ha, J. Kim, and C.-H. Choi, "Trajectory generation and control for a biped robot walking upstairs," *International Journal of Control, Automation and Systems*, vol. 8, no. 2, pp. 339–351, 2010.
- [14] Y.-D. Hong and K.-B. Lee, "Stable Walking of Humanoid Robots Using Vertical Center of Mass and Foot Motions by an Evolutionary Optimized Central Pattern Generator," *International Journal of Advanced Robotic Systems*, vol. 13, no. 1, p. 27, 2016.
- [15] K. Miura, M. Morisawa, F. Kanehiro, S. Kajita, K. Kaneko, and K. Yokoi, "Human-like walking with toe supporting for humanoids," in *IEEE International Conference on Intelligent Robots and Systems*, 2011.
- [16] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for Atlas," *Autonomous Robots*, vol. 40, no. 3, pp. 429–455, 2016.
- [17] J. Carpentier, S. Tonneau, M. Naveau, and O. Stasse, "A Versatile and Efficient Pattern Generator for Generalized Legged Locomotion," *IEEE International Conference on Robotics and Automation*, no. 3, pp. 3555–3561, 2016.
- [18] A.-C. Hildebrandt, M. Demmeler, R. Wittmann, D. Wahrmann, F. Sygulla, D. Rixen, and T. Buschmann, "Real-Time Predictive Kinematic Evaluation and Optimization for Biped Robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016.
- [19] T. Buschmann, S. Lohmeier, M. Bachmayer, H. Ulbrich, and F. Pfeiffer, "A Collocation Method for Real-Time Walking Pattern Generation," in *IEEE/RAS International Conference on Humanoid Robots*, 2007.
- [20] D. Wahrmann, A.-C. Hildebrandt, R. Wittmann, D. Rixen, and T. Buschmann, "Fast Object Approximation for Real-Time 3D Obstacle Avoidance with Biped Robots," in *IEEE International Conference on Advanced Intelligent Mechatronics*, 2016. [Online]. Available: <https://youtu.be/6PLN6B4vSHM?list=PLVAvoOVYkpkmFNNgz8cerCTVvVryz1eh0a%7B%%7D0A>.
- [21] R. Wittmann, A.-C. Hildebrandt, D. Wahrmann, D. Rixen, and T. Buschmann, "Real-Time Nonlinear Model Predictive Footstep Optimization for Biped Robots," *IEEE-RAS International Conference on Humanoid Robots*, pp. 711–717, 2015. [Online]. Available: <https://www.youtube.com/watch?v=46YIWkYWuoY>.
- [22] D. Kraft, "Algorithm 733; TOMP - Fortran modules for optimal control calculations," *ACM Transactions on Mathematical Software*, vol. 20, no. 3, pp. 262–281, 1994.