

Real-Time Predictive Kinematic Evaluation and Optimization for Biped Robots

Arne-Christoph Hildebrandt, Manuel Demmeler, Robert Wittmann,
Daniel Wahrmann, Felix Sygulla, Daniel Rixen and Thomas Buschmann

Abstract—Collision-free walking in cluttered environments is still an open issue for humanoids. Most current approaches use heuristics with large safety margins to plan the robot’s motion. That way, the chance of collisions can be greatly reduced but the robot movements are limited artificially. In this context, we extend our framework for motion generation and whole-body collision-avoidance by an online predictive kinematic parameter evaluation and optimization: We propose to evaluate the initial parameter set describing the walking pattern by integrating the full kinematic model of the robot. In the model our local optimization technique for collision avoidance is taken into account. Initial parameter sets, which are kinematically infeasible due to kinematic limits or collisions can be identified and adapted before the motion is executed. Additionally, the parameter set is optimized according to a chosen cost function using a gradient method and the step time is adapted according to a desired mean velocity. The optimization method is applicable to different representations of the walking pattern. The method is presented with simulation results obtained with our multi-body simulation. The method is suitable for real-time control, since the optimization can be stopped if it exceeds a predetermined time budget. In that case, an executable but suboptimal result is used. The proposed procedure is executed before each step which makes it very reactive to changes in the environment or in the user input. We have also validated the real-time performance in experiments with our humanoid *Lola*.

I. INTRODUCTION

Legged robots are able to prove their strength over wheeled robots when discrete footholds are necessary to navigate in complex environments. The robots’ large number of degrees of freedom (DoF) enable them to reach the consecutive footholds using sophisticated motions. Robotic research has come a long way in motion generation in complex environments. However, the exploitation of the full robot’s kinematic potential over a long planning horizon is still an open issue. With today’s processing power, the arising dimensionality problem is difficult to solve in real-time. Recent advances on real-time navigation in cluttered environments reduce the computational cost by solving the motion planning problem in a hierarchical way. [1], [2], [3], [4], among others, propose calculating an optimal footstep plan. Based on the footstep plan the robot’s trajectories are calculated. Most of those approaches either do not exploit the ability of humanoids to step over or onto obstacles or use heuristic approaches to avoid collisions. [2], [4] propose to design collision-free task space trajectories based on a 2.5D map. Although the presented results are very impressive, the

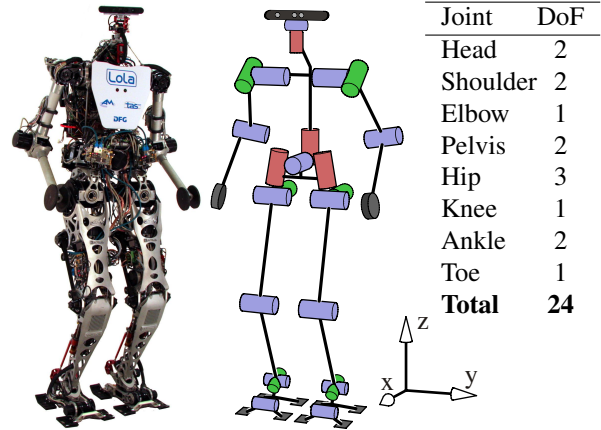


Fig. 1. Photo and kinematic structure of the humanoid robot *Lola*. Joint distribution and used world coordinate system are shown on the right side.

robot’s kinematics are not taken into account. This limits the exploitation of the robot’s potential. [5] extends a footstep planner by dividing the stepping motion into half steps. At first the motion is checked regarding collisions at predefined body configurations. Then, the half steps are smoothed to approach dynamic movements. Although an RRT method is used to handle the high dimensional search space, the footstep planner cycle takes 2.5s, making it difficult for a fast moving robot to react to changes in user input or in the given environment. Another body of literature focuses directly on the stepping motion [6], [7], [8], [9], [10]. The presented methods show remarkable results for the stepping-over-motion when it comes to managing one obstacle at a time. They are however not integrated in a framework involving perception and navigation. Consequently, the methods’ viability has to be proven in more general environments. Regarding our purposes, the published methods mentioned above display several disadvantages: the complex 3D geometries of the robot’s parts and the environment are approximated by line segments or simple boxes. Additionally, potential self-collisions are not taken into account. Hence, the methods only allow movements on a plane. For navigating in an environment with more than one obstacle we expect the need of motions which exploit all of the robot’s DoFs. To solve such issues [11] generates the stepping-over-motion from a whole-body-motion-optimization. However, this is an offline method and not applicable to real-time applications.

Still, treating the stepping motion as a whole-body-motion-optimization problem allows for a more complete exploitation of the robot’s capabilities. In this context, many optimization frameworks for motion planning of different kinds of robotic systems were developed in recent years. [12], [13], among others, present powerful gradient based frameworks, which were also successfully applied on legged robots: [12] shows results of *LittleDog* walking over uneven terrain and [13] applies the presented algorithm on the humanoid *ATLAS* walking statically stable in simulation. However, it is not clear how compatible such frameworks are with our current control architecture. Also, they might not be able to meet our calculation time restrictions and achieving the hard constraints of locomotion of biped robots. Optimization methods similar to our work were reported in [14], [15], [16]. The authors represent trajectories using linear attractor dynamics with a small number of control points to be optimized. This greatly reduces search space and computation time. By optimizing only the control points the search space can be greatly reduced. That way the trajectory optimization is less time-consuming. Since the methods were developed for grasping motions they do not respect the hard timing constraints of humanoids’ locomotion, e.g. foot-ground contact, which are critical in order to maintain balance. In this context, our objective is to further extend our framework for whole-body collision avoidance presented in [17], [18], [19], [20]. In 2014 [18] we proposed a local online optimization technique exploiting all the robot’s swing-foot DoFs for stepping over arbitrarily shaped obstacles while avoiding collisions with the environment and self-collisions. The local optimization technique is combined with an A*-based footstep planner and an on-board vision system in order to close the gap between obstacle recognition, collision-free walking in 2D and whole-body 3D collision avoidance [19], [20]. To improve both robustness and performance of the robot’s movements we propose to replace the previous heuristic approach by evaluating and optimizing the kinematics of the stepping movements. The paper is organized as follows: In Sec. II we give a system overview of the experimental platform used in this work — the robot *Lola* — and a review of our framework for collision avoidance [17], [18], [19], [20]. We then present an online kinematic optimization method for robot movements in cluttered environments in Sec. III. Further, the method is analyzed in simulation and validated in successfully conducted experiments. These results are presented in Sec. IV. Finally, Sec. V is devoted to a conclusion and comments on future work.

II. SYSTEM OVERVIEW

A. Hardware Overview

Our humanoid robot *Lola* weighs approximately 60kg and is 180 cm tall. It has 24 position-controlled joints, which are electrically actuated. A detailed view of the kinematic configuration shows the right hand side of Fig. 1. Note the redundant kinematic structure of the legs with 7 DoF and the pelvis with 2 DoF. For more detailed information see [21], [22]. For environment recognition we set up an

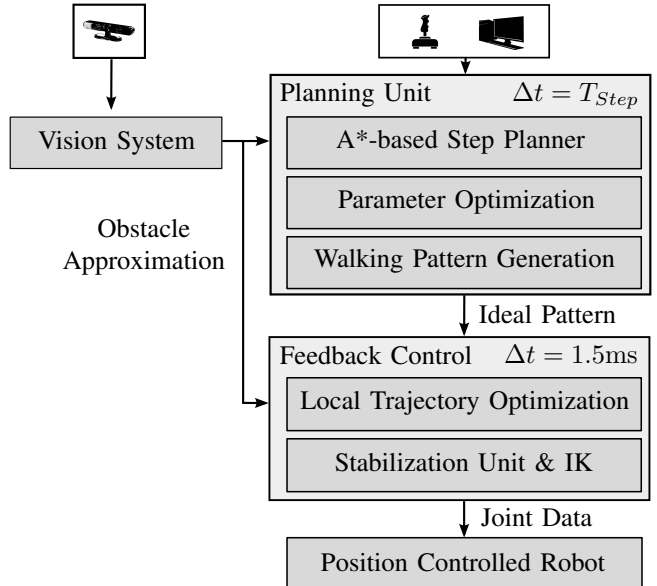


Fig. 2. *Lola*’s real-time walking control system.

Asus Xtion PRO LIVE RGB-D camera¹. It is mounted on a pan/tilt unit on the head of the robot. The vision processing software runs on an on-board computer with an Intel Core i7-4770S @ 3.1 GHz (4x) processor and 8GB RAM on a Linux OS. The control software runs on a computer with the same specification, but on a QNX-RTOS. Both computer communicate via Ethernet using UDP.

B. Control Overview

Our control system follows a hierarchical approach (see Fig. 2). Before each step a planning unit is called. An A*-based step-planner calculates a sequence of footsteps based on high-level user input [19]. The footstep positions are used in the walking pattern generation to determine ideal task-space reference trajectories $w_d \in \mathbb{R}^m$ for the next n_{steps} steps. These trajectories include the CoM position, the torso rotations and the feet’s position and orientation. First, the planning unit determines the CoM height and the feet’s position and orientation trajectories based on a spline representation. Then, it calculates the horizontal CoM and the corresponding reference CoP trajectories using a three mass model. The robot’s walking pattern is determined by a parameter set p_{wp} . It can be divided in a subset p_{fh} describing the footholds of the robot, the step time T_{Step} and the remaining subset p_{mv} describing the robot’s movements (e.g. height of center of mass (CoM)). Hence,

$$w_d = w_d(t) = \text{traj}(w_{d,cur}, p_{wp}) \quad (1)$$

with $w_{d,cur}$ as the start value given by the previous step. w_d are used as set points for feedback control, which is marked as *Stabilization Unit* in Fig. 2. The feedback control runs

¹ASUS Xtion PRO LIVE, see http://www.asus.com/Multimedia/Xtion_PRO_LIVE/

in a cycle time $\Delta t = 1.5$ ms. For the most part, a hybrid position/force control [22] modifies $w_d = w_d(t_k)$ and $\dot{w}_d = \dot{w}_d(t_k)$ at $t = t_k$ to stabilize the robot according to sensor data. The methods proposed in [23], [24] are used to solve the inverse kinematics (IK) for the joint space velocities $\dot{q} \in \mathbb{R}^n$ from the modified task-space velocities \dot{w} . In order to exploit the nullspace for *Lola*, an optimization criterion H_y dedicated to self-collision-avoidance, joint-limit-avoidance and angular-momentum-minimization is minimized [17], [25].

C. Collision Avoidance

Our general framework for collision avoidance is based on the representation of the robot's geometry and environment via swept-sphere-volumes (SSV). This representation allows for efficient and accurate geometry-approximations and fast distance calculations. The key point of our approach is the combination of three contributions: (a) The environment is approximated by a *Vision System* which uses only an on-board RGB-D sensor. It works with a cycle time of 30 ms. For this reason changes of the perceived environment, e.g. moving obstacles, can be taken into account while the robot is walking [20]. (b) With each step an *A*-search based Step Planner* calculates a sequence of footholds. This allows the robot to react in real-time to high-level user input or changes in the perceived environment. The footstep planner uses a dynamic approximation of the feet and the lower legs to take into account the robot's kinematics for collision checking. Thus, high safety margins for footholds can be avoided. Additionally, the step planner determines initial swing-foot trajectories for stepping over or swinging by the obstacle by changing the corresponding parameter in p_{mv} [19]. (c) An integration of the *Trajectory Optimization* in the feedback control allows for collision free movements. It optimizes online the 6D trajectories of the swing foot in workspace taking into account self collisions and collisions with the environment [18]. The combination of the three contributions enables the biped robot to navigate among obstacles taking into account self-collisions and collisions with the environment of the entire robot.

III. KINEMATIC EVALUATION AND PARAMETER OPTIMIZATION

A drawback of this approach is, that the full kinematic movement of the robot is not considered before execution of each step. In the current approach the parameter set p_{wp} , which determines the planned trajectories, is set heuristically - the parameters are set without predictive feedback about the kinematic movement. Since our method is a local method, collision avoidance in 6D could result in complex environments to kinematically non-feasible movements. For this reason, we propose to evaluate and optimize p_{wp} by integrating the kinematic model (cf. Fig 1) over one step taking into account the methods for collision avoidance and nullspace optimization [17], [18]. Fig 2 shows the integration of the method in the overall control architecture, as part of the *Parameter Optimization* step. Fig. 3 shows the workflow of the method.

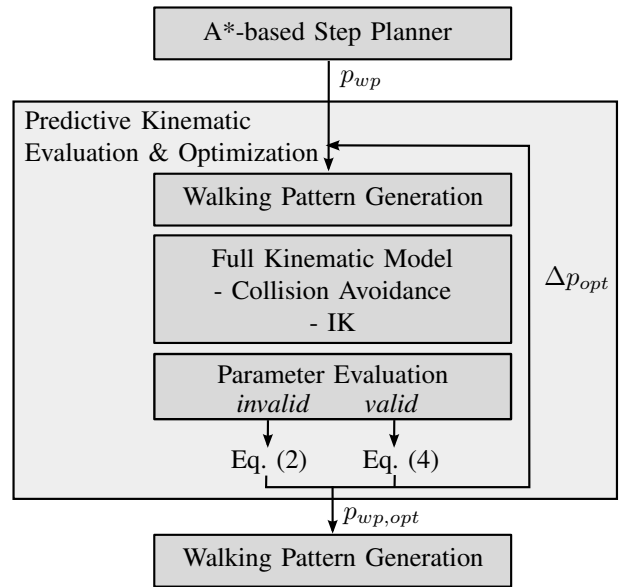


Fig. 3. Predictive Kinematic Evaluation and Optimization.

A. Kinematic Evaluation

Parameter sets which lead to kinematically non-feasible movements can be identified by integrating and analyzing the kinematic movement before the step is executed. Hence, the movement can be replanned using another set of initial parameters including different footholds. We get

$$p_{wp,k+1} = \text{re-init}(p_{wp}) \quad (2)$$

with the re-initialisation of p_{wp} . In this context the challenge is to establish an interaction of the planning modules, step planning and trajectory planning which works reliably and fast enough to meet the real-time requirements.²

B. Parameter Optimization

Additionally to the evaluation of p_{wp} , we propose to optimize p_{mv} along one step. This parameter optimization can be used on-line to improve the walking pattern. The robot's movements are optimized with respect to the following objective function

$$H_{opt}(p_{mv}) = c_q \int_0^{T_{step}} (\dot{q}^T \dot{q} + c_{jl} H_{jl} + c_{coll} H_{coll}) dt. \quad (3)$$

H_{jl} denotes costs produced by joint limit avoidance, H_{coll} denotes costs produced by collision avoidance and c_i denotes weighting factors. The value of $H_{opt} = H_{opt}(p_{mv})$ is evaluated by numerically integrating the robot kinematics (cf. eq. (13)) over one walking step and summing up the values for each integration time-step. In this work the forward Euler integration scheme is used. The updating rule

$$p_{mv,k+1} = p_{mv,k} - h_{opt} \nabla_{p_{mv}} H_{opt} \quad (4)$$

with the updating step-size h_{opt} is used to optimize p_{mv} with respect to H_{opt} after each evaluation k . In this context

²The whole planning process has to be done in less than T_{Step} .

$\nabla_x(\cdot)$ is the $n \times m$ Jacobian of (\cdot) with respect to a variable x . A back-tracking algorithm determines h_{opt} to improve the performance of the optimization process. As long as the Armijo condition

$$H_{opt,k+1} \leq H_{opt,k} - c \|\nabla_{p_{mv}} H_{opt}\|^2 h_{opt} \quad (5)$$

is violated, the updating step-size is decreased by

$$h_{opt} \leftarrow \rho h_{opt} \quad (6)$$

and $p_{mv,k+1}$ is recalculated using eq. (4). The parameters c and $\rho \in (0, 1)$ are freely chosen.

C. Parameter Set

Since the evaluation of H_{opt} for each $p_{mv,k}$ in general and the distance calculation for collision avoidance in particular are time consuming, we focus on a subset p_{opt} of p_{mv} with the following parameters for online application:

- p_{CoM} is referred to as the height of the CoM at the end of each step. The starting and the finishing point are connected via piecewise 5th order polynomials in order to ensure continuity of the CoM movement.
- p_{SF} is a set of control points $s_k, k = 0, \dots, N$ describing the trajectories of the sagittal and horizontal movement of the swing foot.

Foot Trajectory Representation: To include p_{SF} in the parameter optimization, we depend on a trajectory representation which meets the requirements of humanoid walking and which can be described by a variable number of control points. Since collision avoidance depends on the robot's configuration in task-space it seems to be reasonable to set the control points in task-space coordinates on position level. We decided to use a trajectory representation $x(t)$ based on cubic splines $c_k, k = 1, \dots, N$:

$$x(t) = \begin{cases} c_1(t) : & t_0 < t < t_1 \\ c_2(t) : & t_1 < t < t_2 \\ \vdots & \\ c_N(t) : & t_{N-1} < t < t_N \end{cases} \quad (7)$$

with

$$c_k(t) = a_k t^3 + b_k t^2 + c_k t + d_k. \quad (8)$$

a_k, b_k, c_k and d_k are the spline parameters. Cubic splines have the advantage of reducing oscillations, which are more likely with higher dimensional polynomials. For stable humanoid walking the boundary conditions

$$\begin{aligned} x(t_0) &= s_0 & x(t) &= s_N \\ \dot{x}(t_0) &= 0 & \dot{x}(t_N) &= 0 \\ \ddot{x}(t_0) &= 0 & \ddot{x}(t_N) &= 0 \end{aligned}$$

and the continuity on acceleration level of the trajectories are critical. To sum up, these requirements result in $3 \times N$ constraints and $x(t)$ is described by $(N - 1) \times 4$ free parameters. Consequently, we are able to impose for $N - 4$ control points $n_{F,opt}$ additional position constrains and design them according to the desired walking pattern. In

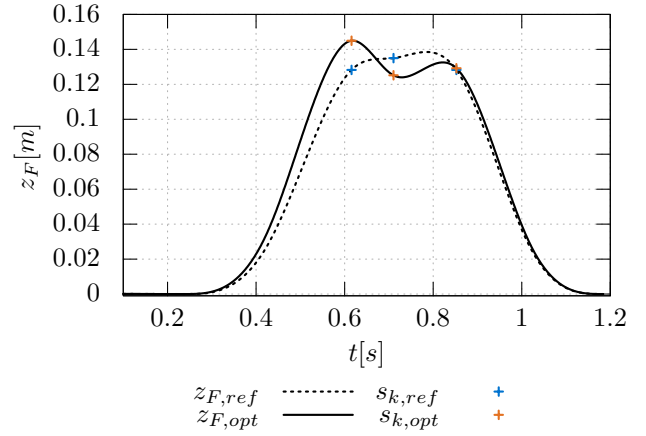


Fig. 4. Vertical foot trajectory of step 8 for Lola stepping over an obstacle before and after optimization.

the current implementation the timing of the control points is fixed. It is chosen heuristically based on simulations and experiments. Fig. 4 shows a trajectory of the vertical foot displacement with $n_{F,opt} = 3$ ($N = 7$) before and after an optimization. In a nutshell, we choose p_{opt} to consist of $n_{F,opt}$ parameters for the vertical and the horizontal swing-foot displacement and the height of the CoM p_{CoM} . Consequently, the optimization space is described by $n_{opt} = 2n_{F,opt} + 1$ optimization parameters in total. This choice of the optimization parameters has proven to be detailed enough to improve the performance of the robot in our experiments. However, the method is open to choose additional parameters or different trajectory representations.

D. Step-Time Adaption

The high level input of our control system may be a desired path, a joystick command or a desired step-parameter set, which can all be presented by a desired velocity v_{des} for each step. The desired walking command is modified by the A*-search to make navigation in cluttered environments possible. In order to guide the robot as close as possible to the desired velocity v_{des} , we adapt T_{step} according to the output of the A*-search, the relative displacement of the next foothold Δl , and v_{des} as follows:

$$T_{Step} = \begin{cases} T_{Step,min} : T_{Step,min} > \frac{\Delta l}{|v_{des}|} \\ \frac{\Delta l}{|v_{des}|} : T_{Step,min} < \frac{\Delta l}{|v_{des}|} < T_{Step,max} \\ T_{Step,max} : T_{Step,max} < \frac{\Delta l}{|v_{des}|} \end{cases} \quad (9)$$

$T_{Step,min}, T_{Step,max}$ are manually set according to the dynamics of the robot.

E. Model

Our method uses the kinematic model of the robot. Sec. II-B summarizes the kinematic structure of our robot and our control framework. The kinematic movements are described by

$$\dot{q} = J_w^\# \dot{w} - (I - J_w^\# J_w) \nabla H_y \quad (10)$$

$$S\dot{w} = S\dot{w}_d - \dot{x}_{coll} \quad (11)$$

with the modification term for collision avoidance

$$\dot{x}_{coll} = [(S^* J_w)^\#]^T \nabla H_{RCA}. \quad (12)$$

∇H_{RCA} is a collision gradient. S, S^* are selection matrices for the relative and absolute swing foot coordinates resp. the remaining ones in the task-space [18]. $J_w, J_w^\#$ are the jacobian and its pseudoinverse, and ∇H_y is the gradient to H_y (cf. sec. II-B). Eq. (10)-(12) is summarized as a first order differential equation of the form

$$\begin{pmatrix} \dot{q} \\ \dot{w} \end{pmatrix} = f(q, w, w_d, \dot{w}_d). \quad (13)$$

The initial conditions are determined by the robot's state at the end of the previous step.

F. Gradient

In the context of real-time application the calculation of the gradient $\nabla_{p_{opt}} H_{opt}$ is challenging: for a small search space of p_{opt} it may be advantageous to determine $\nabla_{p_{opt}} H_{opt}$ numerically. The analytical derivation may take more computational time than the additional integrations for the numerical derivation with finite differences. With growing search space the use of an analytical gradient becomes more significant.

Analytical Gradient: The goal function (3) consists of a sum of functions which depend on w and q . Therefore, we have to differentiate w and q w.r.t. p_{opt} at each time-step of the numerical integration. Similar to [14], differentiation of the discrete form of eq. (13)³

$$\begin{aligned} \begin{pmatrix} q \\ w \end{pmatrix}_{i+1} &= \\ \begin{pmatrix} q \\ w \end{pmatrix}_i &+ \Delta t_i f(q_i, w_i, w_{d_i}(p_{opt}), \dot{w}_{d_i}(p_{opt})), \end{aligned} \quad (14)$$

$$\begin{pmatrix} q \\ w \end{pmatrix}_0 = \begin{pmatrix} q_0 \\ w_0 \end{pmatrix}.$$

delivers the recursive formula for the gradients

$$\begin{aligned} \begin{pmatrix} \nabla_{p_{opt}} q \\ \nabla_{p_{opt}} w \end{pmatrix}_{i+1} &= \begin{pmatrix} \nabla_{p_{opt}} q \\ \nabla_{p_{opt}} w \end{pmatrix}_i \\ &+ \Delta t_i \nabla_q f_i \nabla_{p_{opt}} q_i \\ &+ \Delta t_i \nabla_w f_i \nabla_{p_{opt}} w_i \\ &+ \Delta t_i \nabla_{w_0} f_i \nabla_{p_{opt}} w_{0_i} \\ &+ \Delta t_i \nabla_{\dot{w}_0} f_i \nabla_{p_{opt}} \dot{w}_{0_i}, \end{aligned} \quad (15)$$

$$\begin{pmatrix} \nabla_{p_{opt}} q \\ \nabla_{p_{opt}} w \end{pmatrix}_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

The partial derivatives of f can be calculated analytically. They include derivatives of $J_w, J_w^\#$ and ∇H_{RCA} . The foot trajectories and the CoM trajectories are the p_{opt} -dependent components of w_d . The boundary problem of the CoM movement is solved applying the collocation method as described in [26]. This generates the CoM horizontal

³For the sake of simplicity, we replace $v(t_i) = v_i$.

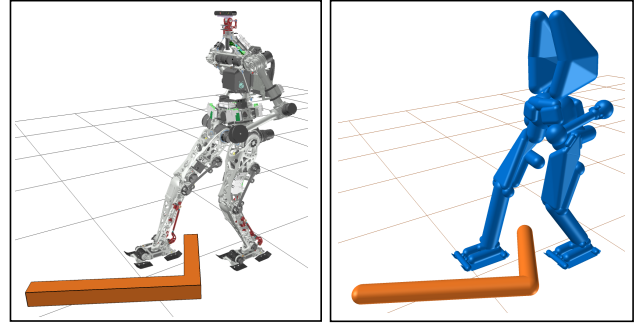


Fig. 5. Left picture: simulation setup, right picture: *Lola's* Collision model in blue and approximation of obstacle in orange

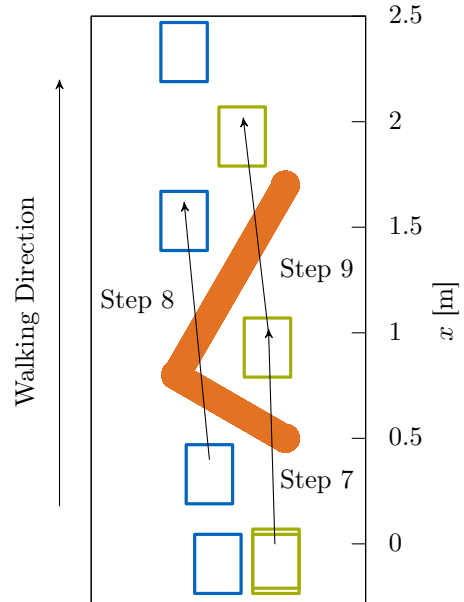


Fig. 6. Step-sequence in simulation testcase. Obstacle in orange, left foot in blue and right foot in green.

reference trajectories. Since an analytical differentiation is computationally expensive and the calculation time of the horizontal CoM reference trajectories is negligible compared to the integration of the robot's state we decided to use numerical gradients for the corresponding derivatives. The derivation of the foot trajectories, as the splines are linearly dependent of p_{opt} , is straight forward.

IV. RESULTS

Simulation

The proposed method was tested in our multibody simulation environment. It takes into account compliant contacts, motor dynamics and the joint control loop. For details see [27]. Due to limitation of space, we present the results of our method in one exemplary environment. Fig. 5 shows the environment of the testcase approximated by our collision-world representation. We chose a testcase which enforced

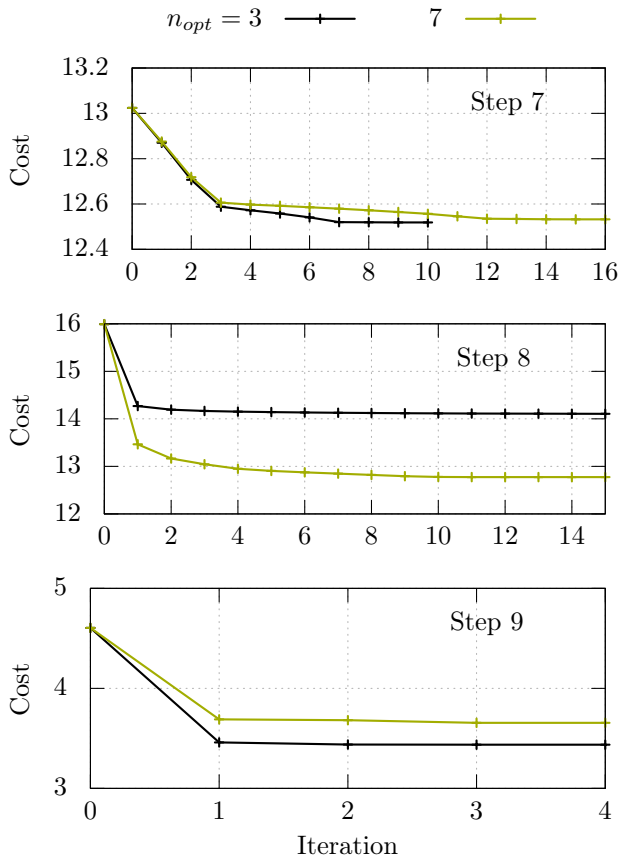


Fig. 7. Costs over iterations for step 7, 8 and 9 in simulation test-case.

a complicated stepping-over-motion⁴ The executed step-sequence, the result of the step-planner, is shown in Fig. 6. The heuristically chosen initial parameter set would lead to a collision between foot and obstacle in step 8. Applying eq. 2 increases the parameter describing the sagittal movement of the swing-foot, and thereby the safety distance between foot and obstacle. This leads to an executable stepping motion. Based on the results of different analyzed stepping-motions we chose a parametrization of the foot trajectories with $n_{F,opt} = 3$. This results in $n_{opt} = 7$ including the CoM height. Fig. 7 shows the calculated costs over iterations. A common result for all steps is the strong cost reduction in the first iterations.

Real-Time Requirements

Our aim is to analyze and optimize each stepping motion before it is executed. Consequently, the planning time for the whole planning process, including the calculation of the footholds and the planning of the desired walking pattern, is limited to $T_{Step} = 0.7...1.2s$. Therefore, we have to introduce a time limit of the optimization which leads to a maximal number of iterations i_{max} for the real-time applica-

⁴The interested reader may compare the testcase with our experiments presented on our *YouTube* - Channel <http://www.youtube.com/appliedmechanicstum>.

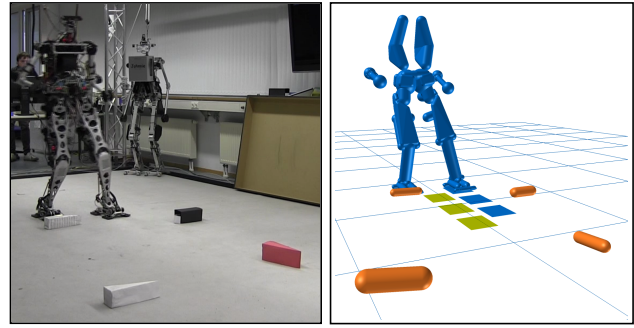


Fig. 8. Left picture: experimental setup, right picture: *Lola*'s Collision model in blue, the approximation of obstacles in field of view in orange and the currently calculated step-sequence (left foot in blue, right foot in green)

tion. In the worst case, the optimization process is not able to converge to a parameter set superior to the initial one. In this case only the kinematic feasibility is checked. Additionally, we analyzed the optimization process to improve the results: (a) We reduced the number of optimization parameters by keeping $N_{F,opt} = 3$, but optimizing only one parameter and shifting the other points of the foot trajectories according to a fixed factor. Consequently, we get $n_{opt} = 3$. Fig. 7 shows that the results for the first iterations are similar to the results with $n_{opt} = 7$. (b) We analyzed different integration step-lengths. The analysis showed that an integration step length which is six times higher than the control cycle time still seems to be an adequate trade-off between accuracy of the kinematic movement of the robot and speed of the integration. Adaptable step lengths depending on the collision gradients showed bad results, which correspond to [28]. The integration of the kinematic model takes on the real-time computer max. $T_{int} = 50ms$ without the gradients, with the gradients $T_{int,grad} = 5T_{int}$.

Experiments

We validated the real-time character of our method in successfully conducted experiments. In the attached video the simulation testcase is shown as well as one example experiment. Fig. 8 shows a snapshot of the experiment and Fig. 11 shows an image sequence from the experiment. The high-level input was $T_{Step} = 0.8s$ and a varying step-length ($l_{Step} = 0...0.3m$). Fig. 9 shows the resulting step-sequence to avoid obstacles. In the attached video of the experiment the step-time adaptations are visible. As described in sec. IV, the used parameter set has a dimension of $n_{opt} = 3$. It consists of dz and dy , describing the vertical resp. sagittal movement of the swing-foot, and H , describing the height of the CoM. Table 10 presents the optimization results based on the initial parameter set which is set by the step-planner. The influence of dy is negligible in our experimental setup, although we assume dy to be important for more complex movements in more complex environments. Since, the robot moves only on flat ground in the experiment, H is optimized to correspond to its maximal limit to reduce the joint velocities in the legs. We suppose the optimal H

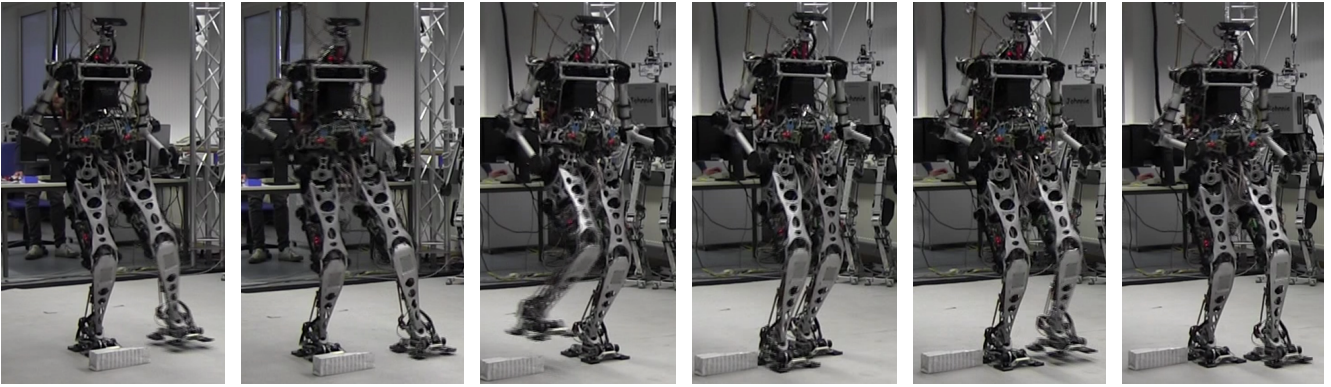


Fig. 11. Walking experiment. *Lola* stepping over obstacles using the presented method in real-time. Image sequence ordered from left to right.

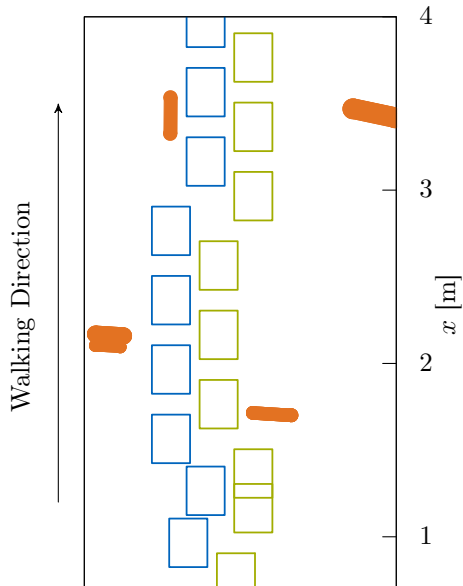


Fig. 9. Step-sequence in experiment. Obstacle in orange, left foot in blue and right foot in green.

Step	Ref. [m]		Opt. [m]		Costs	
	dz	H	dz	H	V_{ref}	V_{opt}
11	0.03	0.88	0.03	0.885	2.2	1.6
12	0.03	0.88	0.03	0.88	0.22	0.22
13	0.03	0.88	0.03	0.882	0.16	0.14
14	0.03	0.88	0.029	0.885	4.3	3.9
15	0.15	0.88	0.14	0.885	4.1	3.8
16	0.03	0.88	0.03	0.885	0.92	0.64
17	0.03	0.88	0.03	0.885	0.99	0.7
18	0.03	0.88	0.03	0.885	1.1	0.78
19	0.03	0.88	0.03	0.885	1.5	1.2

Fig. 10. Comparison of the parameters set by the step-planner and the optimized parameter; the parameter which describes the saggital swing-foot movement is omitted.

changes when the robot steps on stairs or platforms.

V. SUMMARY

We presented an extension of our framework for motion generation and whole-body-collision avoidance by an online kinematic parameter optimization. Before each step is executed the kinematic movement is analyzed by integrating the robot's full kinematic model. We take into account our methods for collision avoidance. (a) Thus, walking pattern which lead to kinematically infeasible movements or collisions can be identified and avoided. This makes the robot's movements more robust. (b) To further improve the exploitation of the robot's potential we propose reducing the walking pattern representation to a small parameter set. This set of parameters is optimized according to a chosen criteria using a gradient method. Introducing a new parametrization of the foot trajectories, the parameter set can be extended. The optimization is especially beneficial for movements in complicated scenarios. (c) The key of the article is the integration in the real-time application: the whole planning process, including footstep planning and parameter optimization, is done in less than a step duration. The parameter optimization can be aborted at any given time. Thus, the robot is still able to react to changing user input or dynamically changing environments. In future we intend to analyze the effect of more sophisticated CoM trajectories. Furthermore, the method has to be tested in more complicated environments including platforms and stairs to improve the interaction between planning and vision system.

ACKNOWLEDGMENTS

This work is supported by the Deutsche Forschungsgemeinschaft (project BU 2736/1-1). Special thanks go to T. Blume and W. Saoud, who helped implementing these ideas and S. Yousefpour, who helped conducting the experiments.

REFERENCES

- [1] J. Seara, K. Strobl, and G. Schmidt, "Information management for gaze control in vision guided biped walking," in *IEEE/RSJ International Conference on Intelligent Robots and System*, vol. 1, no. October. IEEE, 2002, pp. 31–36.
- [2] J. Chestnutt, Y. Takaokaz, M. Doiz, K. Sugaz, and S. Kagamiy, "Safe adjustment regions for legged locomotion paths," in *IEEE-RAS International Conference on Humanoid Robots*. IEEE, dec 2010, pp. 224–229.
- [3] T. Buschmann, S. Lohmeier, M. Schwienbacher, V. Favot, H. Ulbrich, F. von Hundelshausen, G. Rohe, and H.-J. Wuensche, "Walking in unknown environments A step towards more autonomy," in *IEEE-RAS International Conference on Humanoid Robots*. IEEE, dec 2010, pp. 237–244.
- [4] K. Nishiwaki, J. Chestnutt, and S. Kagami, "Autonomous navigation of a humanoid robot over unknown rough terrain using a laser range sensor," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1251–1262, aug 2012.
- [5] L. Baudouin, N. Perrin, T. Moulard, F. Lamiroux, O. Stasse, and E. Yoshida, "Real-time replanning using 3D environment for humanoid robot," in *IEEE-RAS International Conference on Humanoid Robots*. IEEE, oct 2011, pp. 584–589.
- [6] Y. Guan, K. Yokoi, and K. Tanie, "Feasibility: Can Humanoid Robots Overcome Given Obstacles?" in *IEEE International Conference on Robotics and Automation*, vol. 1, no. April. IEEE, 2005, pp. 1054–1059.
- [7] —, "Stepping over obstacles with humanoid robots," *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 958–973, oct 2006.
- [8] B. Verrelst, O. Stasse, K. Yokoi, and B. Vanderborght, "Dynamically Stepping Over Obstacles by the Humanoid Robot HRP-2," in *IEEE-RAS International Conference on Humanoid Robots*. IEEE, dec 2006, pp. 117–123.
- [9] O. Stasse, B. Verrelst, B. Vanderborght, and K. Yokoi, "Strategies for Humanoid Robots to Dynamically Walk Over Large Obstacles," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 960–967, aug 2009.
- [10] M. Arbulu, A. Kheddar, and E. Yoshida, "An approach of generic solution for humanoid stepping over motion," in *IEEE-RAS International Conference on Humanoid Robots*. IEEE, dec 2010, pp. 474–479.
- [11] K. H. Koch, K. Mombaur, O. Stasse, P. Soueres, K. H. Koch, K. Mombaur, O. Stasse, and P. S. Optimization, "Optimization based exploitation of the ankle elasticity of HRP-2 for overstepping large obstacles," in *IEEE-RAS International Conference on Humanoid Robots*, 2014.
- [12] M. Zucker, N. Ratliff, a. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. a. Bagnell, and S. S. Srinivasa, "CHOMP: Covariant Hamiltonian optimization for motion planning," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1164–1193, 2013.
- [13] J. Schulman, J. Ho, A. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding locally optimal, collision-free trajectories with sequential convex optimization," *Robotics: Science and Systems*, pp. 1–10, 2013.
- [14] M. Toussaint, M. Gienger, and C. Goerick, "Optimization of sequential attractor-based movement for compact behaviour generation," in *IEEE-RAS International Conference on Humanoid Robots*. IEEE, nov 2007, pp. 122–129.
- [15] M. Gienger, M. Toussaint, N. Jetchev, A. Bendig, and C. Goerick, "Optimization of fluent approach and grasp motions," in *IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2008, pp. 111–117.
- [16] C. Sung, M. Muhlig, M. Gienger, and Y. Uno, "Task-dependent distribution and constrained optimization of via-points for smooth robot motions," in *IEEE International Conference on Robotics and Automation*. IEEE, may 2015, pp. 3563–3569.
- [17] M. Schwienbacher, T. Buschmann, S. Lohmeier, V. Favot, and H. Ulbrich, "Self-collision avoidance and angular momentum compensation for a biped humanoid robot," in *IEEE International Conference on Robotics and Automation*. IEEE, may 2011, pp. 581–586.
- [18] A.-C. Hildebrandt, R. Wittmann, D. Wahrmann, A. Ewald, and T. Buschmann, "Real-time 3D collision avoidance for biped robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, sep 2014, pp. 4184–4190.
- [19] 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*, IEEE, Ed., 2015.
- [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 (submitted)," in *IEEE International Conference on Advanced Intelligent Mechatronics*. IEEE, 2016.
- [21] S. Lohmeier, T. Buschmann, and H. Ulbrich, "System Design and Control of Anthropomorphic Walking Robot LOLA," *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 6, pp. 658–666, dec 2009.
- [22] T. Buschmann, V. Favot, S. Lohmeier, M. Schwienbacher, and H. Ulbrich, "Experiments in fast biped walking," in *IEEE International Conference on Mechatronics*. IEEE, apr 2011, pp. 863–868.
- [23] D. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Transactions on Man-Machine Systems*, vol. 10, no. 2, pp. 47–53, 1969.
- [24] A. Liegeois, "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms," in *IEEE Transactions on Systems*, no. 12, 1977, pp. 868–871.
- [25] M. Schwienbacher, "Vertical Angular Momentum Minimization for Biped Robots with Kinematically Redundant Joints," *International Congress of Theoretical and Applied Mechanics*, pp. 8–9, 2012.
- [26] 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*. IEEE, nov 2007, pp. 1–6.
- [27] T. Buschmann, "Simulation and control of biped walking robots," Ph.D. dissertation, 2010.
- [28] J. T. Betts, "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193–207, mar 1998.