

Real-Time 3D Collision Avoidance for Biped Robots

Arne-Christoph Hildebrandt, Robert Wittmann, Daniel Wahrmann, Alexander Ewald and Thomas Buschmann*

Abstract—The ability to avoid collisions is crucial for locomotion in cluttered environments. It is not enough to plan collision-free movements in advance when the environment is dynamic and not precisely known. We developed a new method which generates locally optimized trajectories online during the feedback control in order to dynamically avoid obstacles. This method successfully combines a local potential field method with a heuristic based on height and width of an obstacle to avoid collisions. The program’s main feature is the integration of obstacles into the framework designed for self-collision avoidance presented in [1] and the collisions avoidance in task-space. We show experimental results validating the method.

I. INTRODUCTION

Research in humanoid robotics is increasingly focusing on the autonomous navigation in cluttered environments. In this context, the ability of legged robots to step over or onto obstacles is mentioned as one of their main advantage over wheeled vehicles.

While [2] proposes a method for autonomous navigation in unknown environments [3], [4], [5], [6], among others, present approaches for autonomous navigation in cluttered environments which are known in advance. They propose global footstep planners to reach goal positions while avoiding obstacles by using the ability to step over or onto obstacles. Although the step-over- or step-onto-motions are used in the proposed algorithms, their focus lies on the generation of a global foot step path and not on the exact trajectories.

Another body of literature tries to close this gap: It deals directly with motion generation for a humanoid stepping over an obstacle. Ref. [7], [8] and [9] investigate the feasibility of humanoid stepping-over-motions. Based on their results they also proposed a quasi-static trajectory planner.

Ref. [10], [11], [12] and [13] shift these results to a dynamical stepping-over-motion. The main change towards enabling a dynamic movement is to take into account the Zero Moment Point (ZMP) feasibility criteria via the preview control proposed by Kajita et al. [14].

In [13] obstacles and the lower part of the swing leg are represented as boxes. Based on this representation their proposed algorithm checks collisions of the robot with the obstacle for several key configurations via distance calculations between the boxes before the step is executed. They further develop smooth swing foot trajectories and verify if this motion is feasible. However, collisions are not checked between key configurations.

Ref. [12], [11] and [10] follow a similar approach. In the step

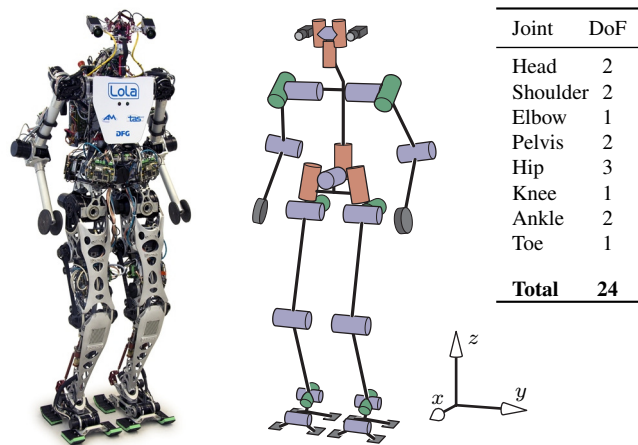


Fig. 1. Photo and kinematic structure of the biped humanoid robot Lola. The right side shows the joint distribution and the used world coordinate system.

planning process they determine the required step length and the appropriated waist height to get a collision-free double support phase. This basis allows the computation of the foot trajectories. One important feature in comparison to [13] is the horizontal online adaptation of the swing foot trajectory for collision avoidance. Unlike [13], their collision detection algorithm relies on a 2D line segment model of the robot’s legs and the obstacle.

Although the presented frameworks show impressive results, they have several disadvantages regarding our purposes. They only consider collisions between the obstacle, represented by a simple box and the swing leg. But they do not include potential self-collisions or the complex 3D geometries of the robot’s parts and the environment. Thus, the frameworks are limited to foot movements in a plane and they don’t allow for more general movements which also include the exploitation of all of the swing foot’s degrees of freedom (DoF) which would improve collision avoidance.

Another research field related to our framework presented in this paper is the field of grasping motions. The proposed solutions already address several problems mentioned above concerning the stepping-over-motion.

The main idea of the work as presented in [15], [16], [17], [18], [19] is to consider the robot motions as an optimization problem. To integrate collision avoidance — that is self collisions or collisions with obstacles — the bodies and obstacles are approximated by bounding capsules. This representation allows for an efficient distance calculation which is used to map distance constraints in the optimization.

*Institute of Applied Mechanics, Technische Universität München, 85748 Garching, Germany. E-mail: arne.hildebrandt@mytum.de

As the proposed frameworks figure as solvers for optimization problems, constraints, such as joint limits or collision avoidance, can be taken into account flexibly.

Treating the trajectory generation as an optimization problem allows for a more flexible collision avoidance as well as for trajectories optimizing chosen objective functions.

Since the methods presented in [17], [18] and [19] solve for the whole trajectory, we cannot use them to generate complex walking motions in real-time. The methods proposed by [15], [16] are less time consuming, but they are tailored to manipulation tasks and do not, e.g., guarantee that the timing constraints for foot-ground contact are exactly satisfied, which is critical in order to maintain balance.

For our work we empathize the publication of Behnisch et al. [18]. They combine a global sampling based search algorithm in task-space with a local potential field based method which adapts the global solution and which we will use in a similar way in this work.

Our objective is to develop a method which generates locally optimized trajectories online in order to avoid obstacles. In this paper we present a new method which combines a local potential field method with a heuristic based on height and width of an obstacle. We add obstacle avoidance in task-space to our framework for self-collision avoidance [1]. This has several advantages:

Our framework enables the robot to avoid precisely modeled obstacles. In addition the local potential field method realizes locally optimized trajectories while exploring all DoFs of the swing foot, considering angle constraints and potential self-collisions. Furthermore, a local obstacle avoidance during the execution of steps is important to react to changes of the sensed environment.

The rest of this paper is organized as follows: In Sec. II we give an overview of the experimental platform used in this work — the robot *Lola*. We present our method for collision avoidance in Sec. III. Sec. IV is dedicated to presenting results from simulation and their validation by successfully conducted experiments. Finally, Sec. V is devoted to a conclusion and to comments on future work.

II. SYSTEM OVERVIEW

The experiments presented in this paper are performed on the biped humanoid *Lola*. It has 24 electrically actuated DoF, weighs approximately 60 kg and is 180 cm tall. Fig. 1 shows a photo and gives a detailed view of the kinematic configuration. In this work, we want to underline the redundant kinematic structure of the legs with 7 DoF and the pelvis with 2 DoF. For a more details see [20].

III. PROPOSED METHOD

As mentioned above, our framework consists of two separate algorithms: (1) a step sequence and heuristic trajectory adaption (SSTA) that modifies foot placement and reference trajectories during the planning phase to approximately avoid obstacles and (2) a reactive 3D collision avoidance (RCA) for reactively avoiding obstacles, which takes the 3D geometries and kinematics into account. Fig. 2 shows the integration of

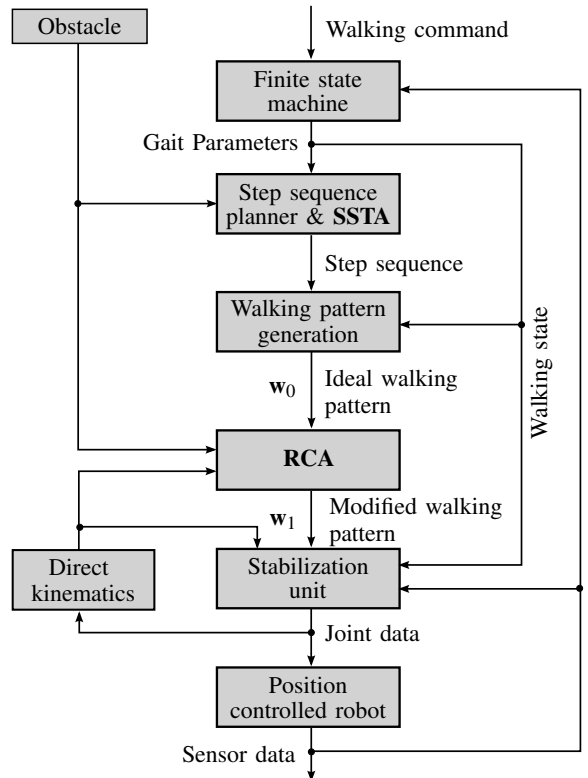


Fig. 2. Integration of the step sequence and heuristic trajectory adaption (SSTA) and the reactive 3D collision avoidance (RCA) in *Lola*'s real-time walking control system. The ideal task-space reference trajectories w_0 and the modified task-space reference trajectories w_1 are emphasized as part of the walking pattern.

both algorithms into the overall planning process. For further information about the walking controller, see [21].

A. Geometric representation

Schwiebacher et al. [1] developed a framework to detect and prevent self-collisions based on distance calculations via swept-sphere-volumes (SSV). It is a computationally efficient method which additionally provides a way to accurately model geometries.

We extended this framework by including obstacles. That way obstacles can be taken into account efficiently in the distance calculation. Currently, however, the obstacles are still added offline by the user.

Our framework consists of the robot R and the environment E . Robot and environment are made up of n_S segments resp. n_O obstacles. Each segment S_i resp. obstacle O_k is approximated by n_{SV} resp. n_{OV} SSV objects V_j . The formulae

$$R := \{S_i : 0 \leq i < n_S\}, \quad (1)$$

$$E := \{O_k : 0 \leq k < n_O\}, \quad (2)$$

$$S_i := \{V_{ij} : 0 \leq j < n_{SV}\}, \quad (3)$$

$$O_k := \{V_{kj} : 0 \leq j < n_{OV}\} \quad (4)$$

describe this mathematically. Note that the environment can be varied dynamically by adding or removing obstacles. According to [1], we define the collision environment C as follows:

C is composed of a set of n_C collision pairs P_l . Each pair consists of either two different segments (S_F, S_T) or a segment and an obstacle (S_F, O_T) :

$$C := \{P_l | 0 \leq l < n_C\}, \quad (5)$$

$$P_l := \{[S_F, S_T | (S_F \wedge S_T \in R) \wedge S_F \neq S_T] \vee [S_F, O_T | S_F \in R \wedge O_T \in E]\}. \quad (6)$$

The framework evaluates the shortest distances d_l between each collision pair. Additionally, it determines the closest pair of points on the related SSV objects. On the right side of Pic. 1 of Fig. 5 the collision model of the robot and a line-SSV as an approximation for a rectangular obstacle are shown.

Since the gradient of the distance function is an essential component of our method, we have to take its mathematical properties into account. Ref. [22] shows that the gradient is discontinuous in some cases which can lead to undesirable behavior if it is included in the robot's control. As a solution they suggest to use strictly convex elements instead of SSVs. However, in our work the exact gradient is not important. Thus we use a filter to smooth the potentially discontinuous influence of the gradient.

The possibility to resort to a highly optimized algorithm is another reason to use the SSVs. By doing so, it is possible to integrate the whole collision model in one control cycle of 2 ms.¹

B. Step sequence and heuristic trajectory adaption (SSTA)

The SSTA is integrated in the hierarchical motion generation process. Based on simple, high level commands such as walking direction and velocity the step sequence planer determines required parameters such as step lengths, step heights and center of mass (CoM) height. From these parameters the walking pattern generator calculates ideal reference trajectories in task-space $\mathbf{w}_0(t) \in \mathbb{R}^m$ resp. ideal task-space velocity trajectories $\dot{\mathbf{w}}_0(t)$. It is called once before a step. For further information concerning the motion generation process see [21].

The SSTA is an extension of the step-sequence-planer presented in [21]. It modifies the stance foot positions and swing foot trajectories to obtain an initial solution according to the size of the obstacle's bounding box. The bounding box is calculated based on the SSV.

In a first step, it discerns whether

- the robot can step on the side of the obstacle with one foot,
- the robot has to overcome the obstacle with both feet (cf. experiments) or
- the obstacle is too large.

Based on this decision the stance foot positions are adapted. The algorithm calculates the stance foot positions in such a way that they are, within a safety margin, in front of and behind the obstacle's bounding box resp. that one stance foot position is on the side of the obstacle's bounding box.

In the next hierarchical level of the motion generation process, the walking pattern generation determines, among others, the six swing foot trajectories. These trajectories are defined as piecewise 5th order polynomials.

In a second step, the SSTA shifts the supporting points of the swing foot trajectories according to the dimensions of the obstacle's bounding box. Depending on the aspect ratio it discerns whether

- the robot has to move the swing foot sideways to the obstacle (cf. experiment with triangular obstacle). If this is the case, the supporting points of the swing foot's tool center point (TCP) lateral trajectory y_F are shifted sideways about the length of the overlap of swing foot and obstacle dy .
- the robot has to move the swing foot over the obstacle (cf. right foot in experiment with rectangular obstacle). Then the height of the supporting points of the swing foot's TCP trajectory z_F are chosen as the height of the obstacle's bounding box.

Since a three mass model is used for calculation of CoM trajectory, the swing foot trajectories are taken into account. Thus, swing foot trajectories can be modified without significantly increasing modeling errors. Additionally, the collocation method proposed by Buschmann et al. [23] would enable modifications of CoM height as yet another parameter. This has, however, not been necessary so far. Since we track the CoM, not the torso position, stretched leg configurations are not a singularity and CoM positions can be controlled via arm joint angles.

Note that the presented modifications don't allow for a collision-free movement.

C. Reactive 3D collision avoidance (RCA)

Our second algorithm is part of the feedback control. The feedback control is based upon the method presented in [21]. It is called in a cycle time dt . In a nutshell, this method modifies $\mathbf{w}_0 = \mathbf{w}_0(t_k)$ resp. $\dot{\mathbf{w}}_0 = \dot{\mathbf{w}}_0(t_k)$ at $t = t_k$ according to sensor data to stabilize the robot. Subsequently, the well known resolved motion rate control method [24] is used to solve the inverse kinematics and the framework proposed by Liégeois [25] is utilized to solve kinematic redundancy. Consequently, we obtain the joint space velocities $\dot{\mathbf{q}} \in \mathbb{R}^n$ from modified task-space velocities $\dot{\mathbf{w}}$ with the Jacobian $J_w = \partial \mathbf{w} / \partial \mathbf{q} \in \mathbb{R}^{m \times n}$ as

$$\dot{\mathbf{q}} = J_w^\# \dot{\mathbf{w}} - (E - J_w^\# J_w) \mathbf{y} \quad (7)$$

Here, E represents the identity matrix and

$$J_w^\# = W^{-1} J_w^T (J_w W^{-1} J_w^T)^{-1} \quad (8)$$

represents the weighted pseudoinverse with an arbitrary diagonal weighting matrix W . For *Lola* the vector \mathbf{y} is

¹Calculations are done under a 32-bit real-time OS QNX 6.5 on an Intel Xeon Quad CPU@2.3 GHz.

a gradient to an optimization criterion H dedicated to self-collision-avoidance, joint-limit-avoidance and angular-momentum-compensation. For further information see [1]. Since \mathbf{y} is only projected into null-space, it doesn't affect the reference trajectories given in task-space. Consequently, ill-chosen reference trajectories can cause self-collisions. In order to face this problem and to integrate obstacle avoidance, we suggest to project a cost function into the task-space.

As mentioned in Sec. I our method uses ideas from [18]. Unlike [18], however, we limit the influence of the cost function to the six DoF of the swing-foot which are described by $\mathbf{x}_F \in \mathbb{R}^6$. Since \mathbf{w} can be written as $\mathbf{w} = [\mathbf{x}_R^T, \mathbf{x}_F^T]^T$ with the remaining coordinates $\mathbf{x}_R \in \mathbb{R}^{(m-6)}$, we define a selection matrix $S \in \mathbb{R}^{6 \times m}$ and a selection matrix $\bar{S} \in \mathbb{R}^{(m-6) \times m}$

$$\mathbf{x}_F = S\mathbf{w}, \quad (9)$$

$$\mathbf{x}_R = \bar{S}\mathbf{w}. \quad (10)$$

Hence, the modified task-space velocity $\dot{\mathbf{w}}_1$ is determined as follows:

$$\dot{\mathbf{w}}_1^T := [\bar{S}\dot{\mathbf{w}}_0^T, \dot{\mathbf{x}}_F^T]^T, \quad (11)$$

$$\dot{\mathbf{x}}_F = S\dot{\mathbf{w}}_0 - (SJ)_w^{\#T} \nabla H. \quad (12)$$

The task-space velocity $\dot{\mathbf{w}}_1$ is now the input of the original feedback control.

The cost function H used above is a combination of objective functions dedicated to collisions and joint limit avoidance. Modifications of \mathbf{x}_F are possible without destabilizing the robot, since the real robot's CoM is taken into account and $m < n$ is chosen (pelvis and arm movements are not planned in the task-space). Hence, a modification of the swing foot trajectories does not influence the CoM trajectory as presented in [26]. Additionally, the null-space is used to reduce the angular-momentum as presented in [1] and [27]. In the attached video, the arm movements to maintain the desired CoM are clearly visible. Starting with the collision avoidance the different parts of H are presented in the following.

1) *Collision avoidance*: Here, we resort to the framework presented in Sec. III-A. According to [1], the cost function for collision avoidance is defined here as a piecewise cubic and quadratic function of d_l :

$$H_{coll}(d_l) = \begin{cases} \frac{s_0}{3(t_c-1)d_a^2} (d_a - d_l)^3 & : d_l < d_a \\ -\frac{s_0}{d_a(t+1)} d_l^2 + s_0 d_l & : d_l < d_a t_c \\ 0 & : d_l \geq d_a \end{cases} \quad (13)$$

where d_a is an activation distance for collision avoidance and s_0 and t_c are parameters that adjust the different objective functions to each other. In this work the parameters d_a , s_0 and t_c are chosen differently for collisions pairs including an obstacle or including only leg segments. For further information about the exact distance calculation or the calculation of the gradient function used in (12) we refer to [1].

2) *Notes*: Because of the time consuming distance calculations in every control cycle the presented method is not applicable to environments with more than a few obstacles. To resolve this problem, we follow a hierarchical approach: Instead of distance calculations of each collision pair including an obstacle, the whole robot is approximated as a line-SSV (*BBL*) and introduced in the distance calculation framework. That way it is possible to verify the distances between *BBL* and obstacles first, and only if the obstacles are inside *BBL*, the other collision pairs are taken into account.

3) *Joint limit avoidance*: In order to avoid joint limits we add a quadratic objective function H_{limit} to take into account the joint limits [1]. Since not all joints (especially of the arms) are not supposed to influence the swing foot movements, we only account for the kinematic chain from the stance foot to the swing foot.

4) *Ideal reference trajectory attractor and re-planning*: For the walking process it is crucial that the robot reaches at the end of each step the ideal reference trajectories $\mathbf{w}_0(t)$. Otherwise an ill-conditioned initial contact of the swing foot with the floor could result in a critical perturbation.

We propose a method composed of two approaches: First, an additional objective function is added to H . It is a quadratic attractor function which relies on the error $\mathbf{e}_F = \mathbf{w}_{1,F} - \mathbf{w}_{0,F}$. H_{att} reduces the influence of the rejecting objective functions defined previously. Therefore it is important to find a parameter configuration which ensures collisions avoidance and joint limits.

Second, the trajectories proposed in [28] are used to plan the error \mathbf{e}_F back to zero. The new ideal reference trajectories of the swing foot $\mathbf{w}_{F,0,n}$ at $t = t_k$ results in

$$\mathbf{w}_{0,F,n} = \mathbf{w}_{0,F} + \mathbf{e}_F(t_k). \quad (14)$$

$\mathbf{e}_F(t)$ denotes the trajectories which lead the error \mathbf{e}_F back to zero. An important characteristic of the trajectories proposed in [28] is that they are overshoot free.

In summary, the cost function H is denoted by

$$H = H_{coll} + H_{limit} + H_{att}. \quad (15)$$

and the re-planning process ensures a correct initial contact of the swing foot with the floor.

IV. RESULTS

The proposed method has been analyzed first in simulation with the framework presented in [29] and then in experiments with *Lola*. Both, simulations and experiments, show the efficiency and the flexibility of the method in application to different sizes and shapes of obstacles during dynamic walking. In the following, we present two experiments. The experiments are performed with (1) a step time of $T_S = 1$ s, (2) a control cycle of $dt = 2$ ms and (3) each step is planned in $10dt = 20$ ms. Only the obstacles are approximated offline by SSVs — all other calculations are done in real-time. Both experiments are performed with the same parameter set.

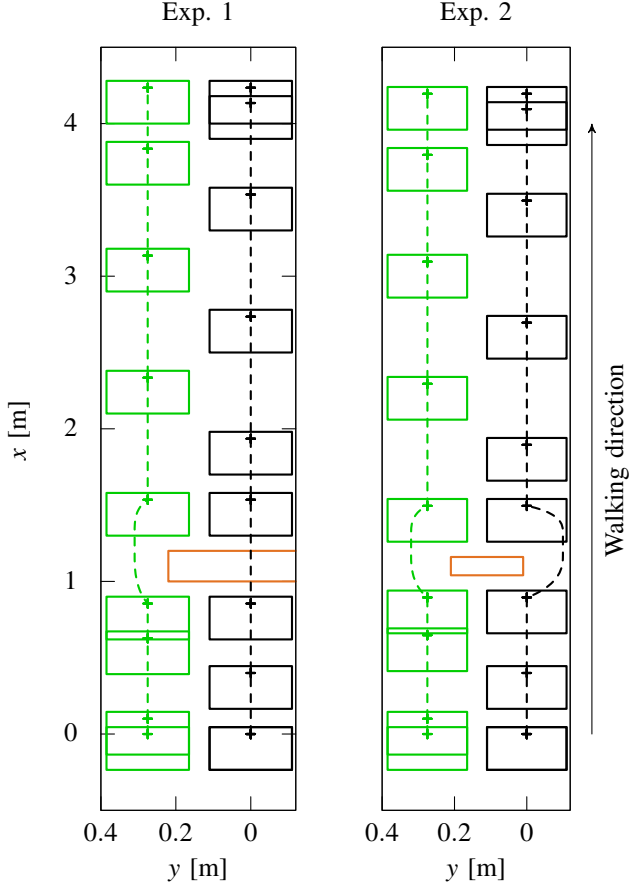


Fig. 3. Modified step sequence: Left stance foot positions shown in green, right stance foot positions in black and obstacle bounding box in red. The dashed lines show the feet's ideal TCP trajectories.

A. Experiment with a rectangular obstacle (Exp. 1)

The first obstacle is a rectangular obstacle which has a height of 16.1 cm, a depth of 8 cm and a width of 35 cm. For the collision avoidance framework it is approximated by a line-SSV with a radius of 10 cm. The line-SSV's center lies 7 cm over the floor.

Fig. 5 shows two snapshots of the attached video with the robot stepping-over (Pic. 1) resp. moving the swing foot sideways (Pic. 2) to the obstacle. The corresponding collision models are shown on the snapshots' right side.

The obstacle's position in the path of the robot would make both feet collide with it. Hence, the SSTA modifies the original step sequence as presented in the following:

- i. The stance foot positions are determined in such a way that the robot stops before the obstacle with a safety margin and the subsequent positions are, with a safety margin, behind the obstacle.
- ii. According to the aspect ratio of height and width of obstacle's bounding box, the robot steps over the obstacle with the right foot and swings the left foot around the obstacle.
- iii. The step height of the right foot's TCP is changed to the height of the obstacle (16.1 cm) and the supporting

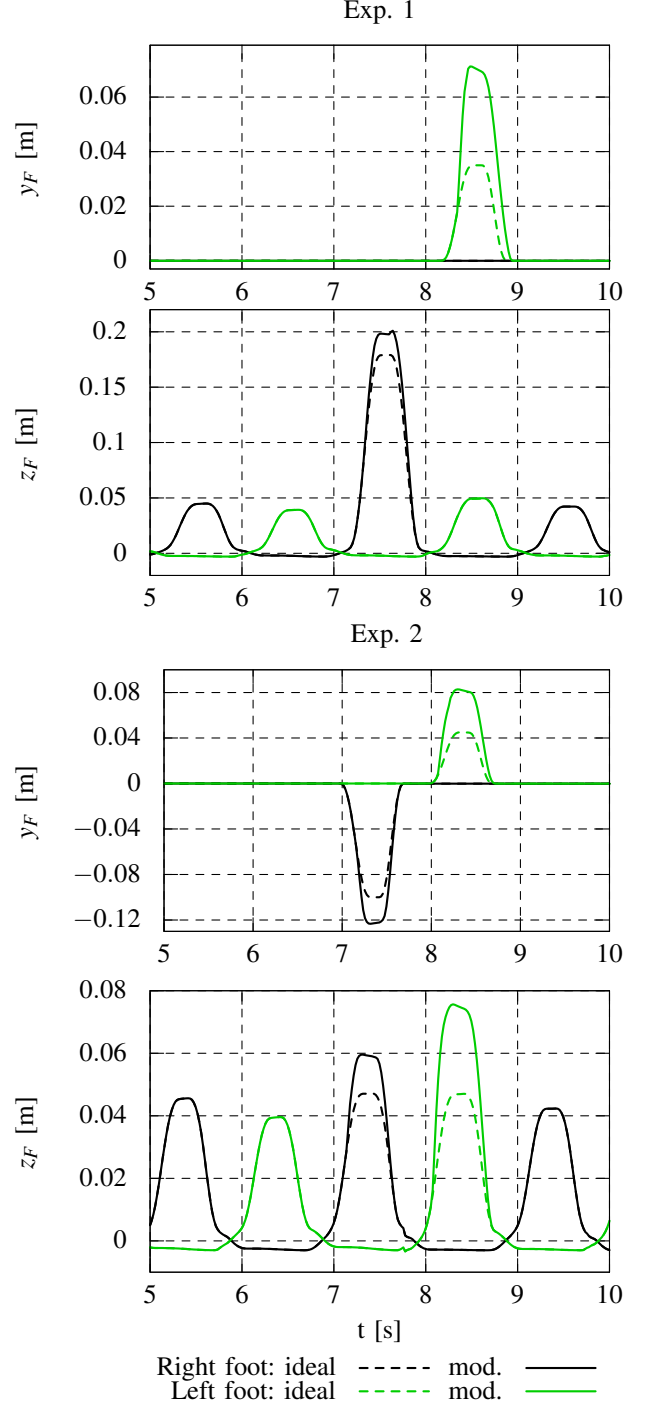


Fig. 4. Vertical trajectories (denoted by z_F) resp. lateral movements (denoted by y_F) of the right (black) and the left (green) foot's TCP. The dashed lines show the ideal trajectories, the solid lines are the trajectories with modifications of the RCA. (Modifications of front and back position and foot orientation not shown).

points describing the left foot's CoM are shifted sideways about the overlap between left foot and obstacle (3.8 cm).

Fig. 3 (Exp. 1), shows the adapted stance foot positions and the adapted foot trajectories resulting from the SSTA. In Fig. 4 the ideal and the modified trajectories of the lateral (denoted by y_F) and the vertical movements (denoted by z_F) of the feet's TCP are shown (Exp. 1).

B. Experiment with a triangular obstacle (Exp. 2)

The second obstacle is a triangular obstacle which has a height of 30 cm, a depth of 9 cm and a width of 13 cm. For the collision avoidance framework it is approximated by a triangle-SSV.

Fig. 5 shows a snapshot of the attached video with the robot moving the right foot sideways to the obstacle (Pic. 3). The corresponding collision model is shown on the snapshot's right side.

The second obstacle is positioned in the robot's path in such a way that both feet would collide with it. Hence, the SSTA modifies the original step sequence as presented in the following.

- i. The stance foot positions are determined in such a way that the robot stops before the obstacle with a safety margin and the subsequent positions are, with a safety margin, behind the obstacle.
- ii. According to the aspect ratio of height and width of the obstacle's bounding box, the robot swings both feet around the obstacle sideways.
- iii. Similar to the first experiment, the supporting points describing the left foot's TCP are shifted to the left about 4.1 cm and the supporting points describing the right foot's TCP are shifted to the right about 10 cm.

Fig. 3 (Exp. 2), shows the adapted stance foot positions and the adapted foot trajectories as a result of the SSTA. In Fig. 4 the ideal and the modified trajectories of the lateral (denoted by y_F) and the vertical movements (denoted by z_F) of the feet's TCP are shown (Exp. 2).

Note that we limit the presentation of the influence of the proposed method on the vertical resp. lateral movements of the feet due to limited space. The RCA modified also the other four task-space trajectories describing the feet's movements in the experiments.

V. SUMMARY

We developed a new method which enables a bipedal robot to walk over and around obstacles in a cluttered environment, while avoiding collisions. We extended the framework designed for self-collision avoidance presented in [1], enabling our robot to dynamically integrate obstacles. With this foundation we successfully combined a local potential field method with a heuristic based on height and width of an obstacle. Thus, the robot is able to flexibly overcome arbitrary shaped obstacles using locally optimized 3D trajectories. The efficiency of the proposed method is demonstrated in experiments. Currently, we are working

on the integration of a vision system which automatically approximates real obstacles in real-time by SSVs.

ACKNOWLEDGMENT

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

REFERENCES

- [1] M. Schwienbacher, T. Buschmann, S. Lohmeier, V. Favot, and H. Ulbrich, "Self-collision avoidance and angular momentum compensation for a biped humanoid robot," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, May 2011, pp. 581–586.
- [2] 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 *2010 10th IEEE-RAS International Conference on Humanoid Robots*. IEEE, Dec. 2010, pp. 237–244.
- [3] 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.
- [4] J. Chestnutt, J. Kuffner, K. Nishiwaki, and S. Kagami, "Planning Biped Navigation Strategies in Complex Environments," in *IEEE International Conference on Humanoid Robotics*, no. Humanoids, 2003.
- [5] J. Chestnutt, Y. Takaokaz, M. Doiz, K. Sugaz, and S. Kagami, "Safe adjustment regions for legged locomotion paths," in *2010 10th IEEE-RAS International Conference on Humanoid Robots*. IEEE, Dec. 2010, pp. 224–229.
- [6] L. Baudouin, N. Perrin, T. Moulard, F. Lamiroux, O. Stasse, and E. Yoshida, "Real-time replanning using 3D environment for humanoid robot," in *2011 11th IEEE-RAS International Conference on Humanoid Robots*. IEEE, Oct. 2011, pp. 584–589.
- [7] Y. Goan, K. Yokoi, and K. Tanie, "Feasibility of humanoid robots stepping over obstacles," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 1, no. c. IEEE, 2004, pp. 130–135.
- [8] Y. Guan, K. Yokoi, and K. Tanie, "Feasibility: Can Humanoid Robots Overcome Given Obstacles?" in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, vol. 1, no. April. IEEE, 2005, pp. 1054–1059.
- [9] K. Yokoi and K. Tanie, "Stepping over obstacles with humanoid robots," *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 958–973, Oct. 2006.
- [10] 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.
- [11] B. Verrelst, K. Yokoi, O. Stasse, H. Arisumi, and B. Vanderborght, "Mobility of Humanoid Robots: Stepping over Large Obstacles Dynamically," *2006 International Conference on Mechatronics and Automation*, pp. 1072–1079, June 2006.
- [12] B. Verrelst, O. Stasse, K. Yokoi, and B. Vanderborght, "Dynamically Stepping Over Obstacles by the Humanoid Robot HRP-2," in *2006 6th IEEE-RAS International Conference on Humanoid Robots*. IEEE, Dec. 2006, pp. 117–123.
- [13] M. Arbulu, A. Kheddar, and E. Yoshida, "An approach of generic solution for humanoid stepping over motion," in *2010 10th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2010, pp. 474–479.
- [14] 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 *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*. IEEE, 2003, pp. 1620–1626.
- [15] M. Toussaint, M. Gienger, and C. Goerick, "Optimization of sequential attractor-based movement for compact behaviour generation," in *2007 7th IEEE-RAS International Conference on Humanoid Robots*. IEEE, Nov. 2007, pp. 122–129.
- [16] M. Gienger, M. Toussaint, N. Jetchev, A. Bendig, and C. Goerick, "Optimization of fluent approach and grasp motions," *Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots*, pp. 111–117, 2008.

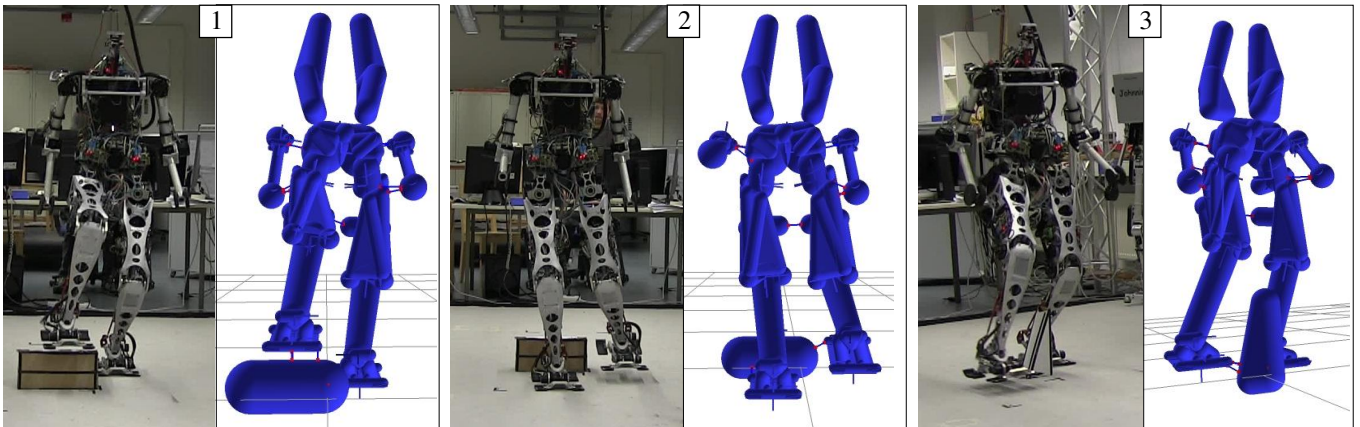


Fig. 5. Snapshots of *Lola* while walking over two different obstacle. The corresponding collision model is shown on the right side of each snapshot.

- [17] M. Behnisch, R. Haschke, and M. Gienger, "Task space motion planning using reactive control," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Oct. 2010, pp. 5934–5940.
- [18] M. Behnisch, R. Haschke, H. Ritter, and M. Gienger, "Deformable trees - exploiting local obstacle avoidance," in *2011 11th IEEE-RAS International Conference on Humanoid Robots*. IEEE, Oct. 2011, pp. 658–663.
- [19] A. El Khoury, F. Lamiroux, and M. Taix, "Optimal motion planning for humanoid robots," in *2013 IEEE International Conference on Robotics and Automation*, vol. 0, no. 2. IEEE, May 2013, pp. 3136–3141.
- [20] 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.
- [21] T. Buschmann, V. Favot, S. Lohmeier, M. Schwienbacher, and H. Ulbrich, "Experiments in fast biped walking," in *2011 IEEE International Conference on Mechatronics*. IEEE, Apr. 2011, pp. 863–868.
- [22] A. Escande, S. Miossec, and A. Kheddar, "Continuous gradient proximity distance for humanoids free-collision optimized-postures," in *2007 7th IEEE-RAS International Conference on Humanoid Robots*. IEEE, Nov. 2007, pp. 188–195.
- [23] T. Buschmann, S. Lohmeier, M. Bachmayer, H. Ulbrich, and F. Pfeiffer, "A collocation method for real-time walking pattern generation," in *2007 7th IEEE-RAS International Conference on Humanoid Robots*. IEEE, Nov. 2007, pp. 1–6.
- [24] D. Whitney, "Resolved motion rate control of manipulators and human prostheses," *Man-Machine Systems, IEEE Transactions on*, vol. 10, no. 2, pp. 47–53, 1969.
- [25] A. Liegeois, "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms," in *IEEE Transactions on Systems*, no. 12, 1977, pp. 868–871.
- [26] T. Buschmann, S. Lohmeier, and H. Ulbrich, "Biped walking control based on hybrid position/force control," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Oct. 2009, pp. 3019–3024.
- [27] M. Schwienbacher, "Vertical Angular Momentum Minimization for Biped Robots with Kinematically Redundant Joints," in *ICTAM*, no. August, 2012, pp. 8–9.
- [28] A. Ewald, J. Mayet, T. Buschmann, and H. Ulbrich, "Generating Smooth Trajectories Free from Overshoot for Humanoid Robot Walking Pattern Replanning," in *Autonomous Mobile Systems*, 2012.
- [29] T. Buschmann, S. Lohmeier, H. Ulbrich, and F. Pfeiffer, "Dynamics simulation for a biped robot: modeling and experimental verification," *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, no. May, pp. 2673–2678, 2006.