

Time-variable, event-based walking control for biped robots

Daniel Wahrmann , Yizhe Wu, Felix Sygulla, Arne-Christoph Hildebrandt, Robert Wittmann, Philipp Seiwald and Daniel Rixen

Abstract

Most walking controllers for biped robots are based on a synchronized phase-based structure, where trajectories are executed following predefined timing constraints. This inherent fixed time dependency makes humanoid robots extremely susceptible to irregularities in terrain compared to their biological counterparts. We present an event-based control strategy which incorporates a time-variable phase to help deal with unexpected early and late contact situations. It results in an improved robustness against such scenarios, as shown by simulation results of our robot *Lola*.

Keywords

Biped robots, locomotion, walking control

Date received: 26 September 2017; accepted: 28 February 2018

Topic: Climbing and Walking Robots

Topic Editor: Manuel Angel Armada Rodriguez

Associate Editor: Elena Lazkano

Introduction

One of the main motivations behind humanoid locomotion research is the ability to navigate in all kinds of terrains. However, humanoid robots are still not ready to work robustly outside laboratory conditions.¹ Some of the factors that make walking control particularly difficult are the high number of degrees of freedom and underactuation due to unilateral contact with the ground. A popular approach to solve this problem consists of using a reduced robot model^{2,3} and analyzing the simplified dynamics between the center mass and the center of pressure, also referred to as zero moment point (ZMP).⁴ Solving these reduced dynamic equations allows one to achieve real-time motion generation. In a nutshell, this is done by first defining foothold positions and timings, and then generating the trajectories for the ZMP and center of gravity (CoG) followed by the inverse kinematics.^{2,5–7} Feedback control is introduced to compensate for modeling errors and external disturbances.^{8–11} In order to walk over uneven terrain, a perception system is introduced, which generates an environment model. Using this model, a search algorithm is used to find

viable footstep locations and then motion planning can continue as before, taking height variation into account via heuristics.^{12–17}

In all works cited above, knowledge about the ground location is assumed and ground contact timing is predefined. Therefore, even small ground location errors have a great effect on this kind of control,¹⁸ especially when compared to their human counterparts.¹⁹ By planning the step durations beforehand, different kinds of errors (e.g. in perception, modeling, or control) may all result in an *early contact* (EC) or a *late contact* (LC) with the ground. Thus, the planned and real walking phases can never be perfectly synchronized. Typically, control and modeling errors as

Department of Mechanical Engineering, Technical University of Munich, Garching, Germany

Corresponding author:

Daniel Wahrmann, Department of Mechanical Engineering, Technical University of Munich, Boltzmannstr. 15, 85748 Garching, Germany.
Email: daniel.wahrmann@tum.de



Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License

(<http://www.creativecommons.org/licenses/by/4.0/>) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (<https://us.sagepub.com/en-us/nam/open-access-at-sage>).

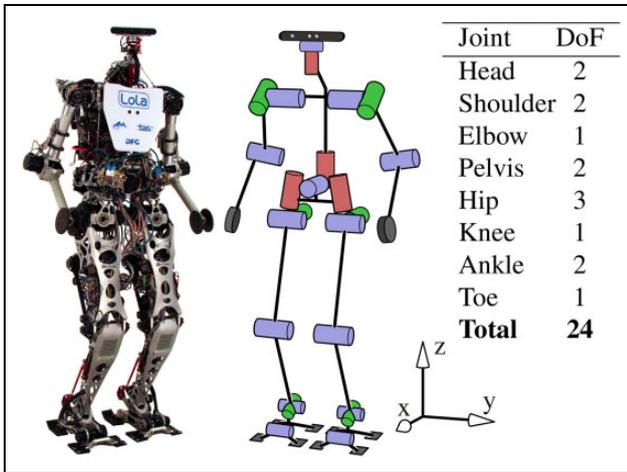


Figure 1. Photo and kinematic structure of the humanoid robot *Lola* with an RGB-D sensor mounted on top.

well as small perception errors in laboratory conditions can be compensated using feedback control. This is possible when the robot's walking is either quasi-static^{12,15} or dynamic at slow moving speeds.^{13,14} Nevertheless, perception errors become larger when performing dynamic walking at higher speeds or even when walking over other kinds of nonrigid, non-flat terrain, such as grass or stones.¹⁸ Some authors presented methods for quick trajectory regeneration^{6,7} during dynamic walking that are based on balance compensation without changing the step duration. Different strategies for online modification of ZMP trajectories were proposed for the HRP-2 robot that allowed it to dynamically walk over carpet tiles, though at low walking speeds.¹¹ Other authors treat these errors as disturbances to be compensated afterwards by quickly modifying future footstep locations, achieving fast dynamic walking.^{20–22}

In contrast, in this article, we propose a flexible walking control that is intrinsically more robust against irregular ground. Instead of fixing the phase durations beforehand, our step duration is variable and depends on sensed contact with the ground, thus making sure that the planned and walking phases are always synchronized. The previous framework for our robot¹⁹ implemented a specific phase to directly deal with detected ECs, which improved the robot's robustness against unexpected obstacles. Now we present a walking controller which deals simultaneously with both EC and LC situations directly at the motion planning stage (in contrast to using feedback control). Our resulting control adapts the walking phase and motion to a direct ground contact detection, improving the robustness of our robot *Lola* (see Figure 1) against irregular terrain.

Other authors have also proposed phase-switching mechanisms for walking control. Some authors presented biologically inspired *central pattern generators* (a review of them can be found in the previous work¹⁹) which have not yet been successfully applied to full-sized humanoid robots as far as the authors know.

A very interesting control strategy for dynamic walking directly depends on the position of the upper body instead of time.²³ This approach was further developed for planar robots,²⁴ where point feet cause an unstable degree of underactuation at the foot–ground contact. By introducing virtual constraints that can be tracked by the position controller, the dynamics of the robot are reduced to its internal dynamics, also called *hybrid zero dynamics* due to the hybrid nature of the biped system.²⁵ Even though the robot's motion does not include statically stable poses, orbital stability of the periodic motion can be achieved by enforcing stable zero dynamics. The authors present a solution which includes inequality constraints in an optimization problem.²⁶ With this strategy, stable dynamic motion over rough terrain was demonstrated with planar robots.²⁷ Although this approach shows an intrinsic robustness against uneven terrain, it is not directly applicable to nonplanar robots²⁸ and is not compatible with existing three-dimensional walking controllers.

In a more related approach, other authors also use contact with the ground as a phase-switch mechanism, but they expect it inside a fixed time window.²⁹ More recently, a robot capable of walking over irregular terrain was presented.³⁰ It shows impressive results but uses a completely different strategy for dynamic walking control based only on controlling the contact forces of the legs against the ground.

In contrast to the methods cited above, we present a method that can be easily implemented in other ZMP-based systems. It allows making the controller more robust against irregular terrain by eliminating the fixed timing restriction of phase switching, using direct ground contact detection instead. The article is organized as follows: In the following section, we give an overview of our walking control, including the previous phase modification strategy¹⁹; later, we focus on our new proposed time-variable phase for LCs; finally, we validate our system using our multi-body simulation system. The article concludes with a discussion of the limitations and future work.

System overview

In this section, we give an overview of our hardware and control framework, followed by the modifications to the walking controller for handling EC and LC scenarios.

Hardware

The *Lola* humanoid robot, as shown in Figure 1, is 1.8 m tall and weighs approximately 60 kg. It has 24 position-controlled, electrically actuated joints. Along with the joint encoders, it has the following additional sensors: an Inertial Measurement Unit (IMU) in the upper body, force–torque sensors in each foot, and contact sensors in the foot soles. A standard RGB-D camera is mounted on top, which is not used throughout this article. The control system runs on

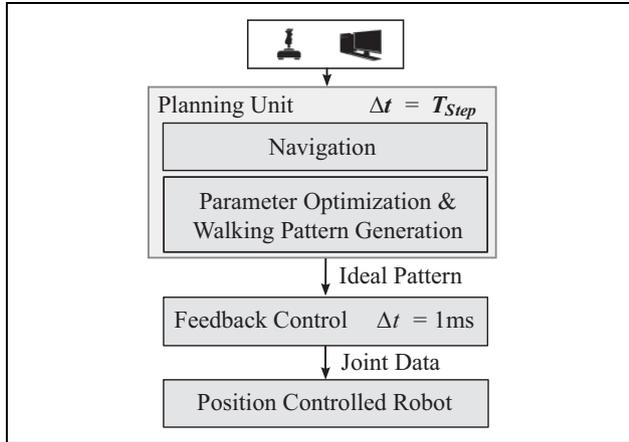


Figure 2. Lola's real-time walking control system.

QNX-RTOS. For more details, see the work of Lohmeier et al.³¹

Control

As explained before, our ZMP-based walking control follows a hierarchical structure which is highlighted in Figure 2. It is mainly divided into a planning unit and a feedback control. The planning unit gets the walking parameters, such as step length and desired velocity, from the user's input. A navigation module takes the walking parameters and collisions with the environment into account to generate footstep locations for the next several steps.³² These are optimized in the parameter optimization module by considering the robot's kinematics.¹⁷ The walking pattern generation module is in charge of generating the robot's ideal trajectories. These are then modified in real time by the feedback control module⁹ for walking stabilization using the IMU and force–torque sensor information.

A state machine is in charge of defining the walking state, which defines the dynamics of the system. The different walking states depend first on the intended action (e.g. start/stop walking) and the planned contact state (e.g. single support (SSP)/double support (DSP)) and are synchronized with predefined timings.³³ These determine the phase of the walking controller and thus the control strategy according to the assumed contact state. For the purpose of this article, we will focus on the two main phases of periodical walking: SSP and DSP (see Figure 3).

Walking pattern generation

We model our robot³ using one mass for the upper body and two small masses at the robot's feet to represent the dynamics of the legs (see Figure 6). The whole system can be written as

$$\mathbf{M}_b \ddot{\mathbf{x}}_b + \mathbf{h} = \mathbf{f} - \mathbf{M}_1 \ddot{\mathbf{x}}_1 - \mathbf{M}_2 \ddot{\mathbf{x}}_2 \quad (1)$$

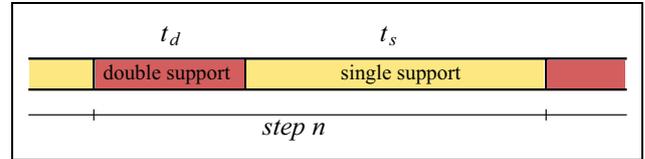


Figure 3. Main walking phases during ideal walking. During one step, the robot transitions between a DSP (duration t_d) and an SSP (duration t_s) contact state. Step duration is as planned. DSP: double support; SSP: single support.

where $\mathbf{f} = (T_y, -T_x, F_z)$ are the ground moments and forces on the robot; \mathbf{h} contains the gravity terms; \mathbf{x}_b , \mathbf{x}_1 , and \mathbf{x}_2 are the positions of the upper body and both feet, respectively; and \mathbf{M}_b , \mathbf{M}_1 , and \mathbf{M}_2 are their corresponding mass matrices. By first defining the trajectories of \mathbf{x}_1 and \mathbf{x}_2 , equation (1) can be used to obtain \mathbf{x}_b . The robot's motion is planned by the walking pattern generation module by sequentially generating the following trajectories³³:

Swing foot: This is parameterized by piecewise quintic polynomials between footstep positions, where it reaches zero velocity and acceleration.

ZMP: It shifts linearly between both feet during the DSP and along one foot during the SSP.

CoG: The trajectory is generated from the ZMP reference trajectory and (1). The boundary value problem is solved by spline collocation.

Load distribution: It calculates the load factors for both feet based on ZMP data.⁹ During the DSP, the load distribution is shifted linearly between both feet.

In our original system,³³ as well as in most humanoid robot controllers, the timings for these phase transitions are fixed (see Figure 3) independently of the exact time the foot touches the ground. In our work, we add two event-based transition phases that react to the sensed ground contact, adapting the walking pattern generation and making the step duration effectively time variable. Changing the step duration results in a displaced CoG: If the step duration is shortened or extended, the CoG will be behind or ahead its planned position with respect to the feet accordingly. In order to compensate for this effect, the following step is adapted accordingly, as explained in the following.

EC response

Previously, an additional *impact* phase was introduced that is activated when a contact with the ground is detected during the SSP.¹⁹ As shown in Figure 4, the state machine switches to the impact phase through this EC event, thus shortening the step's duration

$$t_s^* = t_s - t_i \quad (2)$$

t_s and t_s^* are the planned and modified duration of the SSP, respectively, and t_i the time during which the impact phase

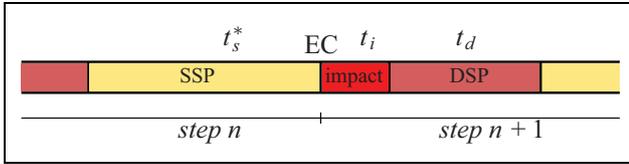


Figure 4. EC response. The step is interrupted and the state machine switches to the impact phase (where the robot is in a DSP state) for the rest of the planned step duration. EC: early contact; DSP: double support.

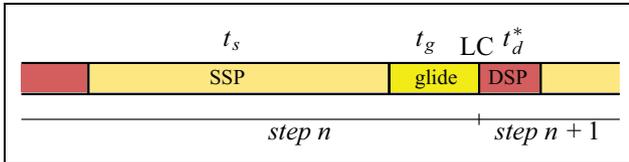


Figure 5. LC response. The step is extended by switching to a gliding phase, until contact with the ground is detected and the DSP can start. LC: late contact; DSP: double support.

is active. After the EC event, the swing foot is immediately stopped, using the “stop trajectory” previously presented.³⁴ During the impact phase, the load distribution is adapted to the new DSP state, and the ZMP and CoG trajectories are unmodified. More details are found in the previous publication.¹⁹

LC response

In this work, we present a walking controller which combines the existing impact phase with a new *glide* phase that specifically targets an LC scenario. This phase deals with an unpredicted, extended SSP state of the robot until a contact with the ground is detected, thus extending the step’s duration as seen in Figure 5. As explained before, the following DSP is shortened to prevent a displaced CoG motion

$$t_d^* = \max(t_d - t_g, t_{\min}) \quad (3)$$

where the duration of the glide phase t_g is the time from the end of the SSP until the contact is detected, t_d and t_d^* are the planned and modified duration of the following DSP, respectively, and t_{\min} is a predefined minimum possible duration of the DSP (in our case one control cycle, or 1 ms). Note that when the condition for t_{\min} is reached the next step is effectively displaced forward in time and not only shortened by t_g . The walking pattern generation t_{\min} process during the gliding phase will be explained in detail in the next section.

Time-variable phase for LC

In an LC scenario, the predefined phase-switch timing results in the planned ZMP making the transition between both feet before the swing foot is in contact with the ground. The robot can tilt over when the ZMP reaches the limit of the stance foot. Additionally, the inclined state of

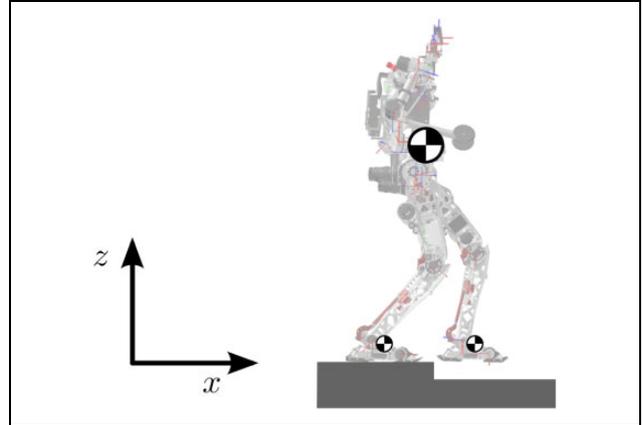


Figure 6. Robot and three mass model at the end of the SSP in an LC scenario—sagittal plane. SSP: single support; LC: late contact.

the robot may cause an initial contact with the toe, causing further destabilization.³³ Our strategy in this scenario becomes keeping the robot from tilting over and achieving a firm contact situation between the swing foot and the ground, thus maintaining the ability to exert torque against the ground. This translates into keeping the ZMP inside the stance foot and maintaining the upper body in a vertical orientation, obtaining less strained walking. In this section, we explain the behavior of our time-variable glide phase throughout all stages of the walking pattern generation (see the previous section) to achieve this goal. These are the ZMP, swing foot, and CoG trajectory generation as well as force control and a necessary footstep replanning. For the sake of simplicity, we will limit our analysis to the sagittal plane (Figure 6), as it is equivalent for the frontal plane.

ZMP trajectory generation

Similar to the normal DSP, during the glide phase, the ZMP is shifted until the boundary (with a safety margin) of the stance foot (in the forward or x direction)

$$x_P(t) = \min(h_g t + c_g, x_B) \quad (4)$$

for $t \in \text{glide}$, where x_P is the location of the ZMP (the y component stays constant). h_g and c_g are chosen to satisfy continuity with the SSP and x_B is defined as a safety margin from the boundary. Instead of moving linearly between both feet, the ZMP is stopped until contact with the ground is detected, in order to keep the foot against the ground and maintain the ability to exert torque against the ground. When the DSP finally starts (after contact is detected), the ZMP motion is adapted to the new phase duration and continued

$$x_P(t) = h_d t + c_d \quad (5)$$

for $t \in \text{DSP}$, h_d and c_d are chosen such that the location of the swing foot is reached in t_d^* . The y component is analogous.

Swing foot vertical trajectory and load distribution

By the end of the SSP, the ground is expected and the swing foot's vertical motion is stopped. As there is no information on where the ground may be, planning a vertical motion to reach the ground becomes difficult. However, if nothing else is done, the foot then stays in the air instead of reaching the ground. Instead, we decide to keep it stopped and let the force control (which can react to a contact with the ground more quickly) move the foot instead. The load distribution is then shifted between both feet, just as it would be during the DSP. More details can be found in the previous publication.⁹

CoG trajectory generation

By the end of the SSP, the CoG is located ahead of the stance foot. In order to keep the ZMP from leaving the support polygon, the CoG position is integrated further in equation (1). The inputs of the equation (\mathbf{x}_1 and \mathbf{x}_2) correspond to the initially planned trajectories of the ZMP (see equations (4) and (5)) and swing foot (constant). The final trajectories for both the ZMP and CoG are obtained from the approximate solution of equation (1) by spline collocation, as explained in the previous publication.³ This results in an accelerated CoG trajectory in the horizontal direction (intuitively, it can be interpreted as the necessary moment to keep the ZMP away from the boundary), increasing the CoG position and velocity along the forward direction while the stance foot stays fixed to the ground. If the swing foot's position is not modified, this increased CoG position and velocity can quickly result in the robot falling forward on the next step. In order to compensate for this displacement, a horizontal trajectory is introduced for the swing foot.

Swing foot horizontal trajectory

Note that a swing foot horizontal motion can easily be taken into account by (1) while generating the CoG trajectory. For the swing foot's horizontal trajectory, we consider the linear inverted pendulum model³⁵

$$\ddot{x} = \frac{g}{z}(x - x_p) \quad (6)$$

where $(x, z)^T$ is the location of the center of mass and x_p the location of the ZMP. The pendulum's orbital energy can be defined³⁵ as the sum of the kinetic energy and an imaginary potential energy around the pivot point x_p : $E = \frac{1}{2} \left(\dot{x}^2 - \frac{g}{z}(x - x_p)^2 \right)$. To compensate for the CoG velocity, we calculate a foot displacement such that the pendulum's orbital energy is maintained (using the one at the end of the SSP as reference)

$$\dot{x}^2 - \frac{g}{z}(x_p - x)^2 = {}_n\dot{x}^2 - \frac{g}{z}({}_n x - {}_n x_p)^2 \quad (7)$$

where ${}_n x$ refers to the value of x at the end of the SSP of step n . The right side of equation (7) is stored at the beginning of the glide phase and x_p is calculated every cycle for the actual values of x and \dot{x} . Note that for the new energy value we consider the pendulum with the swing foot instead of the stance foot as shown in Figure 7. After obtaining x_p , we calculate the new modified swing foot position

$$x_{\text{foot}}^* = x_{\text{foot}} + (x - {}_n x) + k \left((x_p - x) - ({}_n x - {}_n x_p) \right) \quad (8)$$

where x_{foot} and x_{foot}^* are the initial and modified planned footstep location, respectively, and the second term accounts for the CoG displacement (see Figure 7). The third "energy term" consists of the variation of the CoG-ZMP distance with respect to the one at the end of the SSP, in order to obtain a continuous trajectory. We introduce a heuristic factor k to compensate for model errors, such as the unaccounted-for swing-foot motion ($k = 0.75$ in our implementation). Taking the time derivative of equations (8) and (7), the horizontal velocity becomes

$$\dot{x}_{\text{foot}}^* = \dot{x} + \frac{kz}{g} \frac{\ddot{x}}{x_p - x} \quad (9)$$

Footstep replanning

Even though the swing foot horizontal motion helps to reduce the effect of an accelerated CoG, its high speed can still have a destabilizing effect on further steps. If the CoG velocity increases beyond a certain margin, the next footstep location is also modified in order to slow the robot down to the planned walking speed. In our implementation, the footstep replanning is activated when $t_g > 0.2$ s.

Again we use the orbital energy criterion energy and compare the energy at the end of the SSP for both steps

$${}_{n+1}\dot{x}^2 - \frac{g}{z}({}_{n+1}x_p - {}_{n+1}x)^2 = {}_n\dot{x}^2 - \frac{g}{z}({}_n x - {}_n x_p)^2 \quad (10)$$

where we take the planned values of ${}_{n+1}x$ and ${}_{n+1}\dot{x}$. We can again solve for ${}_{n+1}x_p$ and obtain the new footstep location with (8).

Results

We validate our presented control strategy with our multi-body simulation for the *Lola* robot.³³ It handles unilateral and compliant contacts with the ground and takes motor dynamics and control loops into consideration. A video of the simulations can be found at <https://youtu.be/FPpyDLKVICY>.

The first scenario we consider is an unexpected LC. The robot starts walking on a platform which abruptly ends after a few steps. For different heights, we compare simulations for the following: (a) the normal control system (consisting of the previously described walking pattern generation

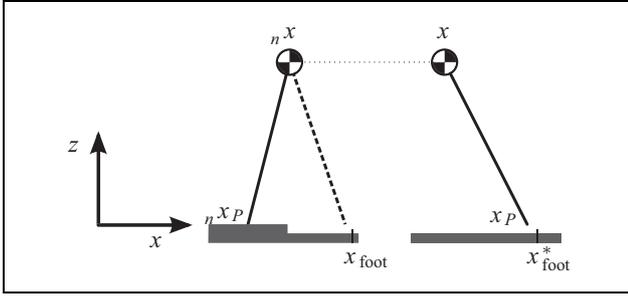


Figure 7. The swing foot horizontal trajectory is generated by analyzing the orbital energy of the linear inverted pendulum model. Left: reference energy at the end of the SSP. Right: an accelerated CoG requires a displaced pivot point to slow it down. SSP: single support; CoG: center of gravity.

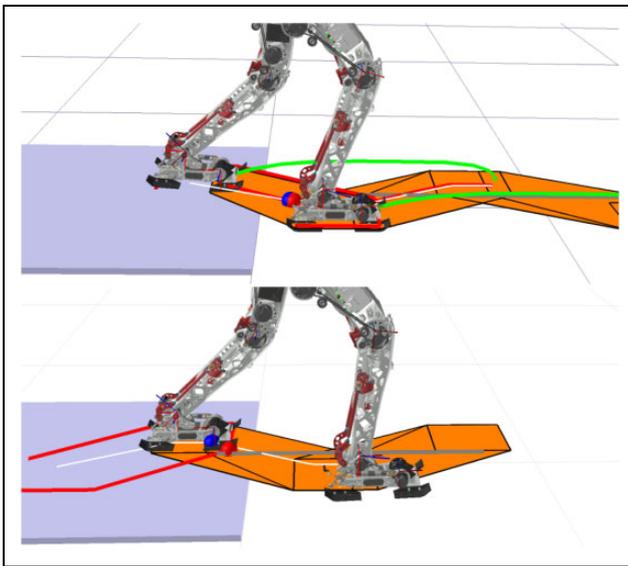


Figure 8. Simulation of an LC scenario (3-cm height in this case), before the swing foot touches the ground. Top: normal control. Bottom: impact and glide phases active. The planned footstep positions, along with the support polygons (orange), are projected on the expected ground. The glide phase keeps the ZMP (red) inside the actual support polygon while accelerating the swing foot and CoG away from their original trajectories. LC: late contact; ZMP: zero moment point; CoG: center of gravity.

strategy without either EC or LC response³³), (b) only the impact phase active,¹⁹ and (c) both impact and glide phases active (this work).

Several simulations with varying parameters were performed. It is confirmed that the impact phase shows a consistent improvement with respect to the normal control even in LC scenarios (as was previously shown¹⁹). The reason behind this is that due to the LC the robot tilts, resulting in an EC for the next step where the impact phase becomes relevant. Additionally, it is observed that adding the glide phase leads to better results than both previous cases, as the effect of the initial LC is reduced before the

Table 1. Robustness.

Control strategy	Maximum platform height (cm)
Normal	3.4
Impact	7.7
Impact+glide	8.7

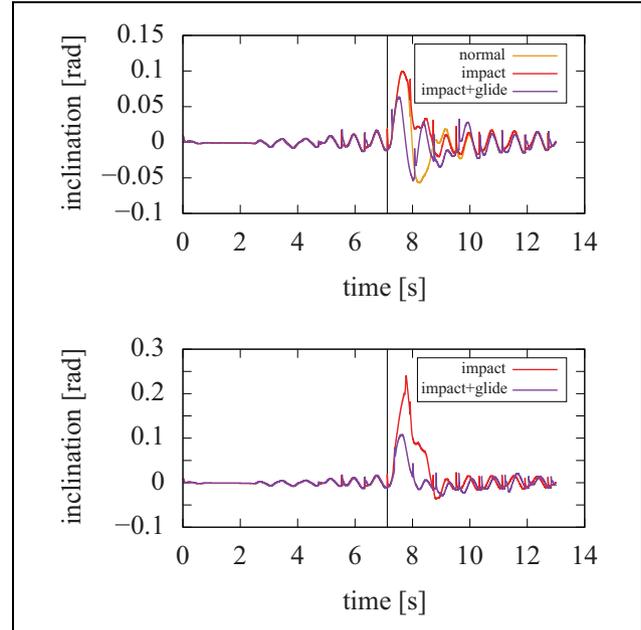


Figure 9. Upper body inclination (around the y axis) for a 3-cm (top) and a 7-cm (bottom) LC scenarios with different control strategies. The vertical line indicates the end of the SSP. The normal control is not shown for the larger height value as it results in the robot tilting over. LC: late contact; SSP: single support.

next step is reached. This behavior is explained in greater detail in the following.

In Figure 8, a screen shot of the normal and “impact+glide” cases for the same platform height is shown. It can be seen how the glide phase helps to maintain the ZMP position and the upper body inclination while the ground has not been reached. In terms of robustness, the robot can also consistently overcome a higher change in the platform height without tilting over. In order to test this, simulations were performed for each control mode by sequentially increasing the platform height by 0.1 cm (starting at ground level) until the robot falls down or a joint limit is reached. The maximum height difference which it can safely overcome can be seen in Table 1.

As discussed in the previous work, we consider the upper body orientation as an indicator of the robot’s stability.¹⁸ In Figure 9, we plot the upper body orientation for two exemplary height values to compare the different cases. The normal control results in the robot tilting over for all height values over 3.4 cm (see Table 1). Therefore, its result is only plotted for the smaller height value, where

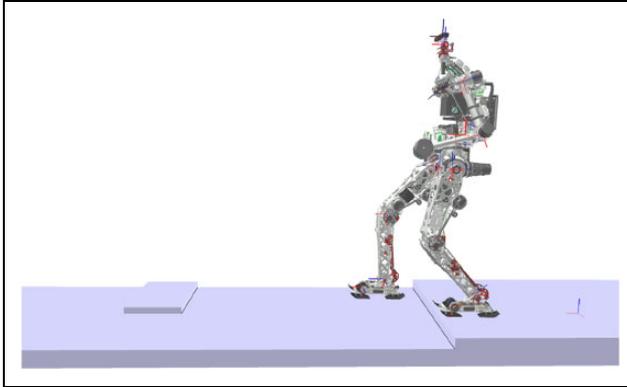


Figure 10. Simulation of complex scenario. A few steps after a 6-cm LC, there is an unexpected 4-cm obstacle (the obstacle location is adapted so that it is directly met by the swing foot in both control cases). LC: late contact.

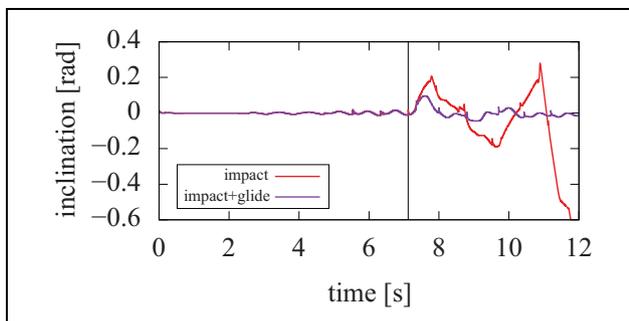


Figure 11. Upper body inclination (around the y axis) with different control strategies for a combined scenario: a 6-cm LC followed by a 4-cm EC. The vertical line indicates the end of the SSP before the LC. The normal control is not shown as it results in the robot tilting over after the LC. Without the glide phase, the disturbance becomes so large that the robot tilts over after the EC. LC: late contact; EC: early contact; SSP: single support.

the normal control and impact phase show consistent behavior during the LC up until the next contact with the ground ($t \approx 8$ s). Afterwards, the impact phase prevents the robot from tilting backwards and maintains a smaller value for the upper body orientation henceforward. In contrast to both previous cases, the glide phase already maintains a low value of the upper body orientation during the LC, preventing high values which could lead to the robot tilting over (this effect can be better appreciated in the bottom plot). The glide phase results in a less pronounced EC by the next step and a less strained walk than both previous cases.

As a final scenario, we consider an unexpected obstacle after the platform, to also analyze how the glide phase performs in an EC situation. A screenshot of the experiment is shown in Figure 10 and the resulting upper body orientation can be seen in Figure 11. Again the normal control is not shown as the robot tilts over. Similarly to Figure 9, the introduction of the glide phase maintains a

low value for the upper body orientation during the LC. This time, it becomes even more relevant as it allows the robot to return to its original walking state more quickly, thus being able to overcome the latter obstacle. Without the glide phase, the robot does not recover in time and tilts over after encountering the obstacle.

Conclusions

In this article, we presented a walking controller for ZMP-based systems that is inherently robust against unexpected irregular terrain. We compliment our previous framework¹⁹ with an extra time-variable phase that specifically deals with LC scenarios. The result is an event-based walking controller with a variable step duration that depends directly on the detected ground contact. It improves the robustness of our robot *Lola* against irregular terrain and perception errors. Future work includes testing on the actual robot in more complicated scenarios. For some particular cases (e.g. unexpected narrow holes), the swing foot trajectory may be counterproductive. One possible solution would be the specific monitoring of horizontal contact (using the force/torque sensors) to better adapt the footstep trajectory. Additionally, a more reactive and comprehensive force control strategy could yield enormous benefits.

Declaration of conflicting interests

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

Funding

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

ORCID iD

Daniel Wahrmann  <http://orcid.org/0000-0002-2562-2004>

References

1. DARPA Robotics Challenge (DRC). <http://archive.darpa.mil/roboticschallenge/> (2017, accessed 29 March 2018).
2. Kajita S and Tani K. Study of dynamic biped locomotion on rugged terrain—derivation and application of the linear inverted pendulum mode. In: *1991 IEEE international conference on robotics and automation*, Sacramento, CA, USA, 9–11 April 1991, pp. 1405–1411. NJ, USA: IEEE; ISBN: 0-8186-2163-X; DOI: 10.1109/ROBOT.1991.131811.
3. Buschmann T, Lohmeier S, Bachmayer M, et al. A collocation method for real-time walking pattern generation. In: *2007 IEEE-RAS international conference on humanoid robots*, Pittsburgh, PA, USA, 29 November–1 December 2007, pp. 1–6. NJ, USA: IEEE; ISBN: 978-1-4244-1861-9.

4. Vukobratovic M and Stepanenko J. On the stability of anthropomorphic systems. *Math Biosci* 1972; 15: 1–37.
5. Loffler K, Gienger M and Pfeiffer F. Model based control of a biped robot. In: *2002 7th international workshop on advanced motion control*, Maribor, Slovenia, 3–5 July 2002, pp. 443–448. NJ, USA: IEEE; ISBN: 0-7803-7479-7.
6. Nishiwaki K and Kagami S. High frequency walking pattern generation based on preview control of ZMP. In: *2006 IEEE-RAS international conference on humanoid robots*, Orlando, FL, USA, 15–19 May 2006, pp. 2667–2672. NJ, USA: IEEE; ISBN: 0-7803-9505-0.
7. Takenaka T, Matsumoto T, and Yoshiike T. Real time motion generation and control for biped robot—1st report: walking gait pattern generation. In: *2009 IEEE/RSJ international conference on intelligent robots and systems*, St. Louis, MO, USA, 10–15 October 2009, pp. 1084–1091. NJ, USA: IEEE; ISBN: 978-1-4244-3803-7.
8. Kajita S, Nagasaki T, Kaneko K, et al. A running controller of humanoid biped HRP-2LR. In: *2005 IEEE international conference on robotics and automation*, Barcelona, Spain, 18–22 April 2005, pp. 616–622. NJ, USA: IEEE; ISBN: 0-7803-8914-X.
9. Buschmann T, Lohmeier S, and Ulbrich H. Biped walking control based on hybrid position/force control. In: *2009 IEEE/RSJ international conference on intelligent robots and systems*, St. Louis, MO, USA, 10–15 October 2009, pp. 3019–3024. NJ, USA: IEEE; ISBN: 978-1-4244-3803-7.
10. Takenaka T, Matsumoto T, Yoshiike T, et al. Real time motion generation and control for biped robot—4th report: integrated balance control. In: *2009 IEEE/RSJ international conference on intelligent robots and systems*, St. Louis, MO, USA, 10–15 October 2009, pp. 1601–1608. NJ, USA: IEEE; ISBN: 978-1-4244-3803-7.
11. Nishiwaki K and Kagami S. Strategies for adjusting the ZMP reference trajectory for maintaining balance in humanoid walking. In: *2010 IEEE International Conference on Robotics and Automation*, Anchorage, AK, USA, 3–7 May 2010, pp. 4230–4236. NJ, USA: IEEE; ISBN: 978-1-4244-5038-1; DOI: 10.1109/ROBOT.2010.5510002.
12. Gutmann JS, Fukuchi M and Fujita M. 3D perception and environment map generation for humanoid robot navigation. *Int J Robot Res* 2008; 27(10): 1117–1134. DOI: 10.1177/0278364908096316.
13. Chestnutt J, Takaoka Y, Suga K, et al. Biped navigation in rough environments using on-board sensing. In: *IEEE/RSJ international conference on intelligent robots and systems*, St. Louis, MO, USA, 10–15 October 2009, pp. 3543–3548. NJ, USA: IEEE; ISBN: 978-1-4244-3803-7; DOI: 10.1109/IROS.2009.5354575.
14. Nishiwaki K, Chestnutt J, and Kagami S. Autonomous navigation of a humanoid robot over unknown rough terrain using a laser range sensor. *Int J Robot Res* 2012; 31(11): 1251–1262. DOI: 10.1177/0278364912455720.
15. Fallon MF, Marion P, Deits R, et al. Continuous humanoid locomotion over uneven terrain using stereo fusion. In: *IEEE-RAS international conference on humanoid robots*, Seoul, South Korea, 3–5 November 2015, pp. 881–888. NJ, USA: IEEE; ISBN: 9781479968855; DOI: 10.1109/HUMANOIDS.2015.7363465.
16. Wahrmann D, Hildebrandt AC, Wittmann R, et al. Fast object approximation for real-time 3D obstacle avoidance with biped robots. In: *2016 IEEE international conference on advanced intelligent mechatronics*, Banff, AB, Canada, 12–15 July 2016, pp. 38–45. NJ, USA: IEEE; ISBN: 9781509020645; DOI: 10.1109/AIM.2016.7576740.
17. Hildebrandt AC, Demmeler M, Wittmann R, et al. Real-time predictive kinematic evaluation and optimization for biped robots. In: *IEEE/RSJ international conference on intelligent robots and systems*, Daejeon, South Korea, 9–14 October 2016, pp. 5789–5796. NJ, USA: IEEE; ISBN: 9781509037612; DOI: 10.1109/IROS.2016.7759852.
18. Wahrmann D, Knopp T, Wittmann R, et al. Modifying the estimated ground height to mitigate error effects on bipedal robot walking. In: *IEEE international conference on advanced intelligent mechatronics*, Munich, Germany, 3–7 July 2017. NJ, USA: IEEE; DOI: 10.1109/AIM.2017.8014226.
19. Buschmann T, Ewald A, Ulbrich H, et al. Event-based walking control—from neurobiology to biped robots. In: *2012 IEEE/RSJ international conference on intelligent robots and systems*, Vilamoura, Portugal, 7–12 October 2012, pp. 1793–1800. NJ, USA: IEEE; ISBN: 978-1-4673-1736-8; DOI: 10.1109/IROS.2012.6385783.
20. Wittmann R, Hildebrandt AC, Wahrmann D, et al. State estimation for biped robots using multibody dynamics. In: *2015 IEEE/RSJ international conference on intelligent robots and systems*, Hamburg, Germany, 28 September–2 October 2015, pp. 2166–2172. NJ, USA: IEEE; ISBN: 978147999934; DOI: 10.1109/IROS.2015.7353667.
21. Wittmann R, Hildebrandt AC, Wahrmann D, et al. Real-time nonlinear model predictive footstep optimization for biped robots. In: *2015 IEEE-RAS international conference on humanoid robots*, Seoul, South Korea, 3–5 November 2015, pp. 711–717. NJ, USA: IEEE; ISBN: 9781479968855; DOI: 10.1109/HUMANOIDS.2015.7363432.
22. Wittmann R, Hildebrandt AC, Wahrmann D, et al. Model-based predictive bipedal walking stabilization. In: *2016 IEEE-RAS international conference on humanoid robots*, Cancun, Mexico, 15–17 November 2016. NJ, USA: IEEE; ISBN: 9781509047178; DOI: 10.1109/HUMANOIDS.2016.7803353.
23. Pratt J, Chew CM, Torres A, et al. Virtual model control: an intuitive approach for bipedal locomotion. *Int J Robot Res* 2001; 20(2): 129–143. Thousand Oaks, United States: SAGE. DOI: 10.1177/02783640122067309.
24. Chevallereau C, Abba G, Aoustin Y, et al. Rabbit: a testbed for advanced control theory. *IEEE Control Syst Magaz* 2003; 23(5): 57–79. DOI: 10.1109/MCS.2003.1234651.
25. Westervelt E, Grizzle J, and Koditschek D. Hybrid zero dynamics of planar biped walkers. *IEEE Trans Autom Control* 2003; 48(1): 42–56. DOI: 10.1109/TAC.2002.806653.
26. Westervelt E, Chevallereau C, Morris B, et al. *Feedback Control of Dynamic Bipedal Robot Locomotion (Automation*

- and control engineering*), Boca Raton, Florida, USA, Vol. 1. CRC Press, 2007. ISBN: 1-42005-372-8.
27. Sreenath K, Park HW, Poulakakis I, et al. A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on MABEL. *Int J Robot Res* 2011; 30(9): 1170–1193. DOI: 10.1177/0278364910379882.
 28. Buss B, Ramezani A, Akbari Hamed K, et al. Preliminary walking experiments with underactuated 3D bipedal robot MARLO. In: *2014 IEEE/RSJ international conference on intelligent robots and systems*, Chicago, IL, USA, 14–18 September 2014, pp. 2529–2536. NJ, USA: IEEE; DOI: 10.1109/IROS.2014.6942907.
 29. Morisawa M, Kanehiro F, Kaneko K, et al. Reactive biped walking control for a collision of a swinging foot on uneven terrain. In: *2011 IEEE-RAS international conference on humanoid robotics*, Bled, Slovenia, 26–28 October 2011, pp. 768–773. ISBN: 9781612848686; DOI: 10.1109/Humanoids.2011.6100885.
 30. Hubicki C, Grimes J, Jones M, et al. ATRIAS: design and validation of a tether-free 3D-capable spring-mass bipedal robot. *Int J Robot Res* 2016; 35(12): 1497–1521. DOI: 10.1177/0278364916648388.
 31. Lohmeier S, Buschmann T, and Ulbrich H. System design and control of anthropomorphic walking robot LOLA. *IEEE/ASME Trans Mech* 2009; 14(6): 658–666. DOI: 10.1109/TMECH.2009.2032079.
 32. Hildebrandt AC, Wahrmann D, Wittmann R, et al. Real-time pattern generation among obstacles for biped robots. In: *2015 IEEE/RSJ international conference on intelligent robots and systems*, Hamburg, Germany, 28 September–2 October 2015, pp. 2780–2786. NJ, USA: IEEE; ISBN 9781479999941; DOI: 10.1109/IROS.2015.7353759.
 33. Buschmann T. *Simulation and control of biped walking robots*. Munich: Dissertation, Technische Universität München, 2010.
 34. Ewald A, Mayet J, Buschmann T, et al. Generating smooth trajectories free from overshoot for humanoid robot walking pattern replanning. *Auton Mobile Syst* 2012; 2012: 89–97. DOI: 10.1007/978-3-642-32217-4_10.
 35. Kajita S, Kobayashit A, and Tani K. Dynamic walking control of a biped robot along a potential energy conserving orbit. In: *1990 IEEE/RSJ international conference on intelligent robots and systems*, Ibaraki, Japan, 3–6 July 1990, pp. 789–794. NJ, USA: IEEE; DOI: 10.1109/IROS.1990.262497.