

Biologically Inspired Deadbeat Control for Running: From Human Analysis to Humanoid Control and Back

Johannes Engelsberger, Paweł Kozłowski, Christian Ott, and Alin Albu-Schäffer

Abstract—This paper works toward bridging the gap between observations and analysis of human-running motions, i.e., motion science and robust humanoid robot control. It is based on the concept of biologically inspired deadbeat (BID) control, which facilitates both 3D running on flat ground and on 3D stepping stones. Further contributions include explicit foot step targeting during running, leg crossover avoidance, and the embedding of BID control into a quadratic-program-based whole-body controller. The controller is based on the encoding of leg forces and center-of-mass (CoM) trajectories during stance as polynomial splines, allowing for intuitive and purely analytical controller design. It allows a real-time implementation, is highly robust against perturbations, and enables versatile running patterns. This paper provides a method for purely analytical foot-step targeting, introduces a new method to increase kinematic feasibility on complex robot models, and presents advanced whole-body running simulations, including high-speed running and push recovery. The paper closes the circle to human motion science by comparing BID-based CoM trajectories and ground reaction forces to data from human-running experiments.

Index Terms—Biologically inspired, bipedal robots, deadbeat, human running, running control, stepping stones.

I. INTRODUCTION

BIOLOGICAL forms of locomotion—such as human walking and running—have evolved over millions of years. They are the product of relentless selection and can, thus, to some extent be regarded as optimal for traversing natural environments. The analysis and decoding of natural locomotion poses a complex yet exciting field of research for biomechanics researchers. Their results can serve as inspiration and object of comparison for roboticists. From an engineering point

of view, gaited forms of locomotion—once fully understood—promise highly increased mobility of machines as compared with wheel-based locomotion. Overcoming a set of stepping stones, as shown in Fig. 1, is one possible example in which a legged robot may have advantages over other machines of similar size and weight.

The first efforts in robotic bipedal locomotion have been put in the subdomain of bipedal walking. Over the decades, the field of bipedal walking control has made major progress. Alongside successes in passive dynamic walking [3], one of the major breakthroughs has been the introduction of zero-moment point control [4], [5] for bipedal walking. More recently, several successful walking control algorithms have been presented, e.g., [6]–[11], to name but a few. Recently, bipedal walking algorithms have reached a level that is close to actual application in real-world scenarios [12]. Most walking algorithms attempt to keep the robot in a fully actuated state, which facilitates the use of standard control methods.

In contrast, during flight, some of the robot's states are unavoidably underactuated, which makes running and hopping challenging tasks. Running provides a number of assets, such as high achievable speed and efficiency. Back in 1985, Raibert [13] presented his controller that decomposes running into three parts: vertical hopping dynamics, forward velocity, and attitude control. The controller design is rather heuristic yet very powerful. Aside from few exceptions, such as [14]–[17], most running algorithms are based on the spring-loaded inverted pendulum (SLIP) [18]. Dadashzadeh, Vejdani, and Hurst [19] present a SLIP-based two-level controller for running simulations of the ATRIAS robot. Carver, Cowan, and Guckenheimer [20] show that the number of required recovery steps depends on the goals of the control mechanism and present a SLIP-based controller for two-step recovery using synergies. Vejdani, Blum, Daley, and Hurst [21] introduce bioinspired swing leg control for running on ground with unexpected height disturbances. Wu and Geyer [22] present a deadbeat controller for the 3-D SLIP model that can cope with unknown ground height variations of up to 30% of the leg length. It is based on multidimensional lookup tables and achieves deadbeat control of apex height and heading direction. Yet, since their model assumes energy conservation, the method cannot handle dissipative losses (e.g., during impact). Koepl and Hurst [23] control the stance phase impulse of a planar SLIP model and achieve robust running. Recently, Wensing and Orin [24], [25] computed periodic trajectories of the 3-D SLIP offline and applied a linearized control law to stabilize the virtual SLIP model around the periodic solutions.

Manuscript received January 29, 2016; accepted May 2, 2016. Date of publication August 1, 2016; date of current version August 18, 2016. This paper was recommended for publication by Editor A. Kheddar and Guest Editors K. Yamane and D. Kulic after evaluation of the reviewers' comments. This work was supported in part by the Initiative and Networking Fund of Helmholtz Association through a Helmholtz Young Investigators Group under Grant VH-NG-808.

The authors are with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Wessling, Germany (e-mail: johannes.engelsberger@dlr.de; p.d.kozlowski@gmail.com; christian.ott@dlr.de; Alin.Albu-Schaeffer@dlr.de).

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the author. The material consists of a video, viewable with both Windows Media Player and VLC Player, showing: 1) a bipedal point-mass pointfoot robot running over three-dimensional stepping stones (with state feedback); 2) the humanoid robot Toro running in OpenHRP simulation at up to 5 m/s; and 3) Toro being disturbed by a frontal force of 150 N for 1 s and by a lateral force of 80 N for 1 s: two video sequences for each, one showing BID preview and stickman and the other showing the OpenHRP view.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TRO.2016.2581199

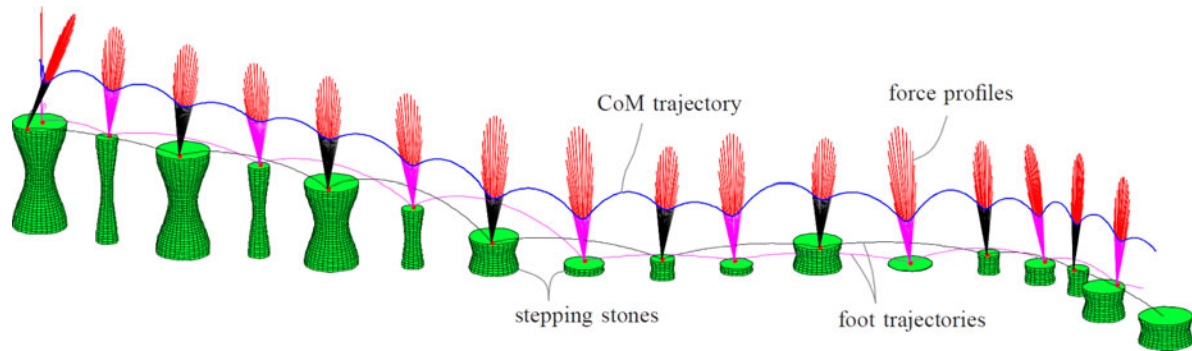


Fig. 1. Bipedal point-mass model running on 3-D stepping stones based on BID control.

The desired leg forces are passed to a whole-body controller, and bipedal running of a simulated humanoid robot is achieved. The method requires offline computation of each desired periodic SLIP gait (including particular turn rates) to obtain the required lookup tables, and the linearized SLIP feed-back controller is only capable of limited acceleration rates. Recently, Park, Park, and Kim [26] presented quadrupedal galloping with the MIT Cheetah 2 based on impulse control. They used third-order Bezier polynomials to encode the leg force profiles. Yet, their method is nominally unstable and designed for constant speeds such that heuristic PD control laws have to be applied to achieve stability and speed control.

Several drawbacks of the previously mentioned works were eliminated in [1]. Here, we proposed the so-called biologically inspired deadbeat (BID) controller that is real-time capable, enables versatile running motions, and is very robust against external perturbations. Dissipative losses are not considered explicitly but are compensated for by the deadbeat controller. It has been inspired by observations from human-running experiments (see Fig. 2) and uses polynomial splines to encode the robot's center-of-mass (CoM) motion and leg forces during stance. The control design is very intuitive and comprehensible. Different running speeds and transitions between them are handled in a clean way. The next two upcoming foot aim points on the ground (i.e., the left and the right one) are predicted at all times, which facilitates the design of appropriate foot trajectories.

One disadvantage of [1] was that the foot positions could not be controlled directly, which caused the danger of leg crossover (especially when running in sharp turns). In [2], we extended the original method to achieve precise foot placement and running on 3-D stepping stones (see Fig. 1). The precise foot placement now enables explicit leg crossover avoidance. Another contribution was the embedding of our running controller into a quadratic program (QP)-based whole-body controller.

The main motivation of this paper is to provide a concept that can act as a bridge between the disciplines of humanoid running research and human motion science. BID control provides a powerful tool for finding and stabilizing diverse running motions. Its embedding into a whole-body controller allows for robust humanoid running simulations, which may serve as a basic tool for future movement science studies.

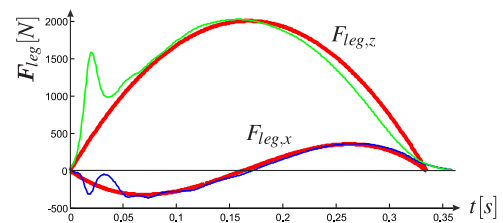


Fig. 2. Comparison of experimentally measured human leg forces (blue/green) and polynomial approximations (red).

This paper is organized as follows. Section II motivates the use of polynomial splines via observations from human-running experiments. Sections III and IV give a short outline of our planning and control framework and recapitulate the flight dynamics. In Section V, the vertical and horizontal boundary conditions are solved, which facilitate the design of our feedback controller presented in Section VI. Sections VII and VIII describe methods to increase force-related and kinematic feasibility. Section IX presents point-mass and whole-body running simulations. Sections X–XII compare the BID control outputs to human experiments, discuss the proposed controller's assets and limitations, and conclude the paper, respectively.

II. HUMAN-RUNNING EXPERIMENTS AS MOTIVATION

The main idea in this paper is to design desired CoM trajectories that produce approximately natural ground reaction forces (GRF) profiles while fulfilling several boundary conditions. It is well known that some physical template models, such as the SLIP, generate GRF similar to the ones observed in human running. Back in 1985, Raibert stated in his book “Closed form expressions relating forward foot placement to net forward acceleration for the one-legged machine are not known” [13]. The lack of closed-form solutions, e.g., for the SLIP, motivates us to find an alternative way of encoding the leg force (F_{leg} , equivalent to GRF). Fig. 2 shows a typical GRF profile recorded during a human-running experiment via force plate. Except for the impact phenomenon at the beginning and the lower slope in the end of stance, the human GRF profiles can be approximated quite well by polynomials of order 2 in the vertical direction and of order 3 in the x -direction. Therefore, our original idea was to approximate the leg force profile during stance via

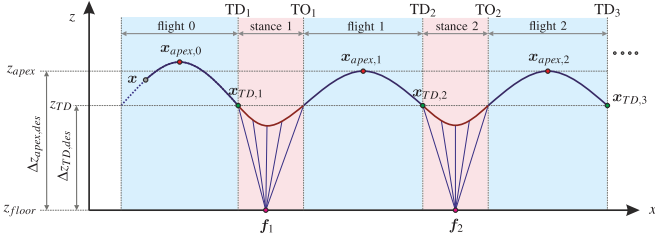


Fig. 3. Preview of upcoming flight and stance phases (planar sketch)—used for design of boundary conditions.

polynomials [1]. The total force \mathbf{F}_{CoM} acting on the CoM can be computed from the leg force \mathbf{F}_{leg} and the gravitational force \mathbf{F}_g via

$$\mathbf{F}_{\text{CoM}} = \mathbf{F}_{\text{leg}} + \mathbf{F}_g = \mathbf{F}_{\text{leg}} + m \mathbf{g}. \quad (1)$$

Here, m is the robot's total mass, and $\mathbf{g} = [0 \ 0 \ -g]^T$ denotes the gravitational acceleration vector. The constant offset between \mathbf{F}_{CoM} and \mathbf{F}_{leg} in (1) and Newton's second law (CoM acceleration $\ddot{\mathbf{x}} = \frac{\mathbf{F}_{\text{CoM}}}{m}$) motivate us to use—during stance—a fourth-order polynomial to encode the vertical CoM position z and fifth-order polynomials to encode the horizontal CoM positions x and y , as this correlates to second- and third-order polynomials for the CoM accelerations \ddot{x} , \ddot{y} , \ddot{z} and thus leg forces. This polynomial encoding can be written as

$$\begin{bmatrix} \sigma(t) \\ \dot{\sigma}(t) \\ \ddot{\sigma}(t) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & t & t^2 & t^3 & t^4 & t^5 \\ 0 & 1 & 2t & 3t^2 & 4t^3 & 5t^4 \\ 0 & 0 & 2 & 6t & 12t^2 & 20t^3 \end{bmatrix}}_{\begin{bmatrix} \mathbf{t}_\sigma^T(t) \\ \mathbf{t}_{\dot{\sigma}}^T(t) \\ \mathbf{t}_{\ddot{\sigma}}^T(t) \end{bmatrix}} \mathbf{p}_\sigma, \sigma \in \{x, y, z\} \quad (2)$$

Here, $\mathbf{t}_\sigma^T(t)$, $\mathbf{t}_{\dot{\sigma}}^T(t)$, and $\mathbf{t}_{\ddot{\sigma}}^T(t)$ denote the time-mapping row vectors that, for a given time t , map the polynomial parameter vectors \mathbf{p}_σ to CoM positions $\sigma(t)$, velocities $\dot{\sigma}(t)$, and accelerations $\ddot{\sigma}(t)$. The last elements of the vectors are greyed out to indicate that they are only used for the horizontal directions but not for the vertical one.

III. OUTLINE OF BIOLOGICALLY INSPIRED DEADBEAT CONTROL METHOD

In this paper, we use a preview of at least two upcoming stance and flight phases, as shown in Fig. 3. The desired relative apex and touch-down (TD) heights $\Delta z_{\text{apex,des}}$ and $\Delta z_{\text{TD,des}}$ are used as design parameters. They indicate how high over the floor the apex of each flight curve (i.e., $\dot{z} = 0$) should be and at what CoM height the TD is supposed to happen. $z_{\text{floor},i}$ denotes the height level of the i th step. Another design parameter, used in this study, is the total stance time T_s (it can vary from step to step), whereas the total flight time T_f results from the boundary conditions chosen in Section V-A. To keep track of the current running state, we use a state machine. It switches from flight to stance, if the CoM is below $z_{\text{TD}} = z_{\text{floor},i} + \Delta z_{\text{TD}}$ and the

vertical velocity is negative, and from stance to flight when the total stance time is over. A timer provides the time in stance $t_s \in [0, T_s]$ and the time in flight $t_f \in [0, T_f]$. They are reset at state transitions.

IV. CENTER-OF-MASS DYNAMICS DURING FLIGHT

Running is a locomotion pattern, which employs alternate flight and (single leg supporting) stance phases. During flight, the CoM cannot be controlled, i.e., it follows its natural dynamics (parabolic path through space). For a given time t , the CoM position $\mathbf{x}(t) = [x(t), y(t), z(t)]^T$ and velocity $\dot{\mathbf{x}}(t) = [\dot{x}(t), \dot{y}(t), \dot{z}(t)]^T$ can be computed as

$$\mathbf{x}(t) = \mathbf{x}_0 + \dot{\mathbf{x}}_0 t + \mathbf{g} \frac{t^2}{2}, \quad (3)$$

$$\dot{\mathbf{x}}(t) = \dot{\mathbf{x}}_0 + \mathbf{g}t, \quad (4)$$

where \mathbf{x}_0 and $\dot{\mathbf{x}}_0$ are the initial CoM position and velocity, respectively. One typical task in running control is to achieve a certain apex height. The apex is the highest point in the ballistic flight curve, i.e., vertical CoM velocity $\dot{z}_{\text{apex}} = 0$. Using this condition and the current vertical CoM velocity \dot{z} instead \dot{z}_0 in the third row of (4), we find the current time to apex Δt_{apex} as

$$\Delta t_{\text{apex}} = \frac{\dot{z}}{g}. \quad (5)$$

The remaining time until TD is computed as

$$\Delta t_{\text{TD}} = \Delta t_{\text{apex}} + \sqrt{\Delta t_{\text{apex}}^2 + \frac{2}{g} (z - z_{\text{TD}})}. \quad (6)$$

Here, $z_{\text{TD}} = z_{\text{floor}} + \Delta z_{\text{TD}}$ is the CoM height at which the TD (flight-to-stance transition) is previewed to happen. With (3), (4), and (6), the previewed TD state can be precomputed for any CoM state $[\mathbf{x}, \dot{\mathbf{x}}]$ as

$$\begin{bmatrix} \mathbf{x}_{\text{TD}} \\ \dot{\mathbf{x}}_{\text{TD}} \end{bmatrix} = \begin{bmatrix} \mathbf{x} + \Delta t_{\text{TD}} \dot{\mathbf{x}} + \frac{\Delta t_{\text{TD}}^2}{2} \mathbf{g} \\ \dot{\mathbf{x}} + \Delta t_{\text{TD}} \mathbf{g} \end{bmatrix}. \quad (7)$$

In this study, the relative TD height is computed as

$$\Delta z_{\text{TD}} = \min \left(\Delta z_{\text{TD,des}}, z - z_{\text{floor}} + \frac{\dot{z}^2}{2g} - \Delta_{\text{apex,TD,min}} \right), \quad (8)$$

i.e., nominally, the desired relative TD height $\Delta z_{\text{TD,des}}$ is achieved, while in the case of perturbations, a minimum height difference between apex and TD $\Delta_{\text{apex,TD,min}}$ is guaranteed, and the solution of (6) is assured to be real.

V. METHOD FOR BOUNDARY CONDITION SATISFACTION AS BASIC MODULE FOR DEADBEAT CONTROLLER

A. Vertical Planning and Boundary Conditions

As mentioned above, the vertical CoM trajectory during stance is encoded via a fourth-order polynomial, i.e., it has five polynomial parameters. These can be derived using five boundary conditions. Fig. 3 graphically displays the used preview of upcoming flight and stance sequences and the corresponding boundary conditions. In this study, for each previewed contact

phase, we make use of four linear vertical boundary conditions that can be combined as

$$\underbrace{\begin{bmatrix} z_{\text{TD},i} \\ \dot{z}_{\text{TD},i} \\ -g \\ -g \end{bmatrix}}_{\mathbf{b}_{z,i}} = \underbrace{\begin{bmatrix} t_z^T(0) \\ t_z^T(0) \\ t_z^T(0) \\ t_z^T(T_{s,i}) \end{bmatrix}}_{\mathbf{B}_{z,i}} \mathbf{p}_{z,i}. \quad (9)$$

Here, i denotes the index of the considered step, and $\mathbf{b}_{z,i}$, $\mathbf{B}_{z,i}$, and $\mathbf{p}_{z,i}$ denote the corresponding boundary condition vector, boundary condition mapping matrix, and vertical polynomial parameter vector, respectively. The first two elements in $\mathbf{b}_{z,i}$ imply that CoM position and velocity at the beginning of stance equal the CoM TD state. The other two elements say that the CoM acceleration at beginning and end of stance equals minus gravity, i.e., the vertical leg force is zero. The general solution of the linear system $\mathbf{B}_{z,i} \mathbf{p}_{z,i} = \mathbf{b}_{z,i}$ is

$$\mathbf{p}_{z,i} = \mathbf{B}_{z,i}^T (\mathbf{B}_{z,i} \mathbf{B}_{z,i}^T)^{-1} \mathbf{b}_{z,i} + \mathbf{r}_{z,i} \tilde{p}_{z,i}. \quad (10)$$

The nullspace base vector $\mathbf{r}_{z,i}$ ensures that $\mathbf{B}_{z,i} \mathbf{r}_{z,i} = 0$. The whole (1-D) nullspace of $\mathbf{B}_{z,i}$ is represented by the scalar variable $\tilde{p}_{z,i}$. The vector $\mathbf{r}_{z,i}$ is computed as

$$\mathbf{r}_{z,i} = \begin{bmatrix} -\mathbf{B}_{z,i,\text{square}}^{-1} \mathbf{B}_{z,i,\text{final}} \\ 1 \end{bmatrix}, \quad (11)$$

where $\mathbf{B}_{z,i,\text{final}}$ is the last column in $\mathbf{B}_{z,i}$, while $\mathbf{B}_{z,i,\text{square}}$ consists of all other columns. Equation (9) encodes the four linear previously described vertical boundary conditions. The fifth boundary condition that we aim to fulfill is the apex height $z_{\text{apex},i}$ of the CoM during the i th upcoming flight phase (see Fig. 3). The vertical CoM state of the i th takeoff (at end of i th stance time $T_{s,i}$) can be computed via (2) as

$$z_{\text{TO},i} = t_z^T(T_{s,i}) \mathbf{p}_{z,i} \quad (12)$$

$$\dot{z}_{\text{TO},i} = t_{\dot{z}}^T(T_{s,i}) \mathbf{p}_{z,i}. \quad (13)$$

With (3) and (5), we can compute the i th apex height as

$$z_{\text{apex},i} = z_{\text{TO},i} + \frac{\dot{z}_{\text{TO},i}^2}{2g}. \quad (14)$$

We are looking for a parameter vector $\mathbf{p}_{z,i}$ that will result in the desired apex height $z_{\text{apex},i,\text{des}}$, which can be computed as

$$z_{\text{apex},i,\text{des}} = z_{\text{floor},i+1} + \Delta z_{\text{apex},\text{des}}. \quad (15)$$

Note that here we use the height $z_{\text{floor},i+1}$ of the upcoming step. Inserting (12) and (13) into (14) leads to a quadratic equation in the unknown scalar variable \tilde{p}_z

$$0 = \frac{\mathbf{r}_{z,i}^T t_z^T t_z^T \mathbf{r}_{z,i}}{2g} \tilde{p}_{z,i}^2 + \left(t_z^T \mathbf{r}_{z,i} + \frac{t_z^T \mathbf{p}_{z,i,0} t_z^T \mathbf{r}_{z,i}}{g} \right) \tilde{p}_{z,i} + \frac{(t_z^T \mathbf{p}_{z,i,0})^2}{2g} - z_{\text{apex},i,\text{des}} + t_z^T \mathbf{p}_{z,i,0} \quad (16)$$

It can be shown that the only valid solution to (16) (yielding positive vertical takeoff velocities) is

$$\tilde{p}_{z,i} = \frac{2 \dot{z}_{\text{TO},i} - g T_{s,i} - \Gamma}{4 T_{s,i}^3} \quad (17)$$

$$\Gamma = \sqrt{g(g T_{s,i}^2 - 4 \dot{z}_{\text{TO},i} T_{s,i} + 8 (z_{\text{apex},i,\text{des}} - z_{\text{TD},i}))}$$

Note that, finally, only (11) and (17) are necessary as inputs for (10) to compute polynomial parameters $\mathbf{p}_{z,i}$ for each previewed step that fulfill all desired vertical boundary conditions.

B. Horizontal Planning and Boundary Conditions

In this study, the derivation for the x - and y -component is equivalent. We use the letter χ to indicate horizontal quantities, i.e., $\chi \in \{x, y\}$. We choose, for each previewed contact phase, the following five linear horizontal boundary conditions:

$$\underbrace{\begin{bmatrix} \chi_{\text{TD},i} \\ \dot{\chi}_{\text{TD},i} \\ 0 \\ 0 \\ \chi_{\text{TD},i+1,\text{des}} \end{bmatrix}}_{\mathbf{b}_{\chi,i}} = \underbrace{\begin{bmatrix} t_{\chi}^T(0) \\ t_{\dot{\chi}}^T(0) \\ t_{\chi}^T(0) \\ t_{\chi}^T(T_{s,i}) \\ t_{\chi}^T(T_{s,i}) + T_{f,i} t_{\chi}^T(T_{s,i}) \end{bmatrix}}_{\mathbf{B}_{\chi,i}} \mathbf{p}_{\chi,i}. \quad (18)$$

Here, $\mathbf{b}_{\chi,i}$, $\mathbf{B}_{\chi,i}$, and $\mathbf{p}_{\chi,i}$ denote the horizontal boundary condition vector, boundary condition mapping matrix, and polynomial parameter vector, respectively. As in Section V-A, the first two elements of $\mathbf{b}_{\chi,i}$ imply that the initial CoM state is equal to the CoM TD state. The next two elements assure that initial and final CoM acceleration are zero, i.e., horizontal leg forces are zero. The fifth element, as an intermediate control target, specifies the horizontal CoM TD position $\chi_{\text{TD},i+1,\text{des}}$ of the upcoming step. Since in the case of no perturbations the horizontal velocity during flight is constant, we can propagate the takeoff state to each upcoming TD position via

$$\begin{aligned} \chi_{\text{TD},i+1,\text{des}} &= \chi_{\text{TO},i} + T_{f,i} \dot{\chi}_{\text{TO},i} \\ &= (t_{\chi,i}^T(T_{s,i}) + T_{f,i} t_{\dot{\chi}}^T(T_{s,i})) \mathbf{p}_{\chi,i}. \end{aligned} \quad (19)$$

Here, the i th time of flight $T_{f,i}$ is computed via (5) and (6). Note that $z_{\text{TO},i}$ and $\dot{z}_{\text{TO},i}$ [used as z and \dot{z} in (5) and (6)] are computed from the vertical polynomial parameter vector $\mathbf{p}_{z,i}$. Thus, the vertical boundary conditions are solved before the horizontal ones. The general solution of (18) is

$$\mathbf{p}_{\chi,i} = \underbrace{\mathbf{B}_{\chi,i}^T (\mathbf{B}_{\chi,i} \mathbf{B}_{\chi,i}^T)^{-1} \mathbf{b}_{\chi,i}}_{\mathbf{p}_{\chi,i,0}} + \mathbf{r}_{\chi,i} \tilde{p}_{\chi,i}. \quad (20)$$

The nullspace base vector $\mathbf{r}_{\chi,i}$ is computed via the equivalent of (11). The horizontal directions have one more polynomial parameter than the vertical one, i.e., one more degree of freedom (DOF). This DOF, represented by the scalar $\tilde{p}_{\chi,i}$ in (20), has an effect on the geometry of the leg force rays in space (see Fig. 4). Our goal is to find the value for $\tilde{p}_{\chi,i}$, which produces the best possible focusing of leg forces, such that these are best feasible for finite-sized (or even point) feet. To this end, for each previewed step, we compute the time-dependent intersection

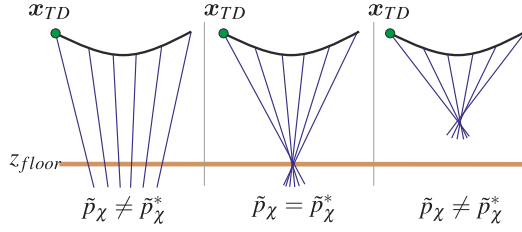


Fig. 4. Effect of \tilde{p}_χ on force ray focusing (lines of action).

point $\mathbf{x}_{\text{int},i} = [x_{\text{int},i}, y_{\text{int},i}, z_{\text{floor},i}]$ of the leg force with the floor and minimize the integral of the mean square deviation from its mean value $\bar{\mathbf{x}}_{\text{int},i}$. For a given time in the i th stance $t_s \in [0, T_{s,i}]$, its horizontal components are

$$\begin{aligned} \chi_{\text{int},i}(t_s) &= \chi(t_s) - \frac{f_{\text{leg},\chi,i}(t_s)}{f_{\text{leg},z,i}(t_s)} (z(t_s) - z_{\text{floor},i}) \\ &= \underbrace{\left(\mathbf{t}_\chi^T(t_s) - \frac{(\mathbf{t}_z^T(t_s) \mathbf{p}_{z,i} - z_{\text{floor},i}) \mathbf{t}_\chi^T(t_s)}{\mathbf{t}_z^T(t_s) \mathbf{p}_{z,i} + g} \right)}_{\mathbf{d}_{\chi,i}^T(t_s)} \mathbf{p}_{\chi,i}. \end{aligned} \quad (21)$$

Here, $f_{\text{leg},\chi,i}(t_s)$ and $f_{\text{leg},z,i}(t_s)$ are the horizontal and vertical components of the leg force $\mathbf{F}_{\text{leg},i}$, and $z(t_s)$ is the height of the CoM. The explicit solution to (10) is given by

$$\mathbf{p}_{z,i} = \left[z_{\text{TD},i}, \dot{z}_{\text{TD},i}, -\frac{g}{2}, -2T_{s,i} \tilde{p}_{z,i}, \tilde{p}_{z,i} \right]^T. \quad (22)$$

Remember that the scalar nullspace variables $\tilde{p}_{z,i}$ are given in (17). Although the horizontal polynomial parameter vectors $\mathbf{p}_{\chi,i}$ are not determined yet, using the third and fourth lines of (18) as constraints, they can be expressed as

$$\mathbf{p}_{\chi,i} = \begin{bmatrix} p_{\chi,i,1} \\ p_{\chi,i,2} \\ 0 \\ -2T_{s,i} p_{\chi,i,5} - \frac{10T_{s,i}^2}{3} p_{\chi,i,6} \\ p_{\chi,i,5} \\ p_{\chi,i,6} \end{bmatrix}. \quad (23)$$

Using (2) and (22), the term $\mathbf{d}_{\chi,i}^T(t_s)$ in (21) is expanded to

$$\mathbf{d}_{\chi,i}^T(t_s) = [1, t_s, \square, t_s^3, t_s^4, t_s^5] + \beta(t_s) \underbrace{\frac{[0, 0, \square, 6t_s, 12t_s^2, 20t_s^3]}{12\tilde{p}_{z,i}t_s(T_{s,i} - t_s)}}_{\mathbf{f}_{\chi,i}^T(t_s)}, \quad (24)$$

where $\beta(t_s) = z_{\text{TD},i} - z_{\text{floor},i} + \dot{z}_{\text{TD},i} t_s - \frac{gt_s^2}{2} - 2\tilde{p}_{z,i}T_{s,i}t_s^3 + \tilde{p}_{z,i}t_s^4$. The \square symbol indicates that the corresponding term has no effect on the product $\mathbf{d}_{\chi,i}^T(t_s) \mathbf{p}_{\chi,i}$ due to the 0 in the third element of $\mathbf{p}_{\chi,i}$ and is thus not considered. Now, cancelling $2t_s$ yields $\mathbf{f}_{\chi,i}^T(t_s) = \frac{1}{6\tilde{p}_{z,i}(T_{s,i} - t_s)} [0, 0, \square, 3, 6t_s, 10t_s^2]$. When the product $\mathbf{f}_{\chi,i}^T(t_s) \mathbf{p}_{\chi,i}$ is evaluated, we can cancel the term

$(T_{s,i} - t_s)$ as well, and the product is equivalently written as

$$\mathbf{f}_{\chi,i}^T(t_s) \mathbf{p}_{\chi,i} = \underbrace{\left[0, 0, 0, 0, \frac{-1}{\tilde{p}_{z,i}}, \frac{-5}{3\tilde{p}_{z,i}}(t_s + T_{s,i}) \right]}_{\mathbf{f}_{\chi,i}^T(t_s)} \mathbf{p}_{\chi,i}. \quad (25)$$

Setting $\mathbf{f}_{\chi,i}^T(t_s) = \check{\mathbf{f}}_{\chi,i}^T(t_s)$ and $\square = 0$ in (24), we find

$$\mathbf{d}_{\chi,i} = \left[1, t_s, 0, t_s^3, t_s^4 + \frac{-\beta(t_s)}{\tilde{p}_{z,i}}, t_s^5 + \frac{-5\beta(t_s)}{3\tilde{p}_{z,i}}(t_s + T_{s,i}) \right]^T. \quad (26)$$

Now, the horizontal components of the mean intersection point $\bar{\mathbf{x}}_{\text{int},i} = [\bar{x}_{\text{int},i}, \bar{y}_{\text{int},i}, z_{\text{floor},i}]$ can be computed via

$$\bar{\chi}_{\text{int},i} = \frac{1}{T_{s,i}} \int_{t_s=0}^{T_{s,i}} \chi_{\text{int},i}(t_s) dt_s = \frac{1}{T_{s,i}} \int_{t_s=0}^{T_{s,i}} \underbrace{\mathbf{d}_{\chi,i}^T(t_s)}_{\mathbf{e}_{\chi,i}^T} dt_s \mathbf{p}_{\chi,i}. \quad (27)$$

Here, $\mathbf{e}_{\chi,i}$ is a constant vector. The simple form of (26) as compared with (24) facilitates the integration, which yields the following analytical solution for $\mathbf{e}_{\chi,i}$:

$$\mathbf{e}_{\chi,i} = \begin{bmatrix} \frac{1}{T_{s,i}} \\ \frac{T_{s,i}}{2} \\ 0 \\ \frac{T_{s,i}^3}{4} \\ \frac{z_{\text{floor},i} - z_{\text{TD},i} - \frac{\dot{z}_{\text{TD},i}T_{s,i}}{2} + \frac{gT_{s,i}^2}{6}}{\tilde{p}_{z,i}} + \frac{T_{s,i}^4}{2} \\ \frac{5T_{s,i}}{2\tilde{p}_{z,i}}(z_{\text{floor},i} - z_{\text{TD},i} - \frac{5\dot{z}_{\text{TD},i}T_{s,i}}{9} + \frac{7gT_{s,i}^2}{36}) + \frac{19T_{s,i}^5}{18} \end{bmatrix} \quad (28)$$

where $\tilde{p}_{z,i}$ is the scalar variable from (17). The deviation of the i th time-varying intersection point from its mean value is

$$\Delta\chi_{\text{int},i}(t_s) = \chi_{\text{int},i}(t_s) - \bar{\chi}_{\text{int},i} = \underbrace{(\mathbf{d}_{\chi,i}^T(t_s) - \mathbf{e}_{\chi,i}^T)}_{\mathbf{k}_{\chi,i}^T(t_s)} \mathbf{p}_{\chi,i}. \quad (29)$$

The square of the deviation at a given time t_s is

$$\Delta\chi_{\text{int},i}^2(t_s) = \mathbf{p}_{\chi,i}^T \underbrace{\mathbf{k}_{\chi,i}(t_s) \mathbf{k}_{\chi,i}^T(t_s)}_{\mathbf{L}_{\chi,i}(t_s)} \mathbf{p}_{\chi,i}. \quad (30)$$

In order to obtain the mean square of the deviation $\chi_{\text{int},i,\text{ms}}$, we once again integrate and insert (20) to achieve

$$\begin{aligned}\chi_{\text{int},i,\text{ms}} &= \mathbf{p}_{\chi,i}^T \underbrace{\frac{1}{T_{s,i}} \int_{t_s=0}^{T_{s,i}} \mathbf{L}_{\chi,i}(t_s) dt_s}_{\mathbf{M}_{\chi,i}} \mathbf{p}_{\chi,i} \\ &= \mathbf{r}_{\chi,i}^T \mathbf{M}_{\chi,i} \mathbf{r}_{\chi,i} \tilde{p}_{\chi,i}^2 \\ &\quad + 2 \mathbf{r}_{\chi,i}^T \mathbf{M}_{\chi,i} \mathbf{p}_{\chi,i,0} \tilde{p}_{\chi,i} \\ &\quad + \mathbf{p}_{\chi,i,0}^T \mathbf{M}_{\chi,i} \mathbf{p}_{\chi,i,0}.\end{aligned}\quad (31)$$

The analytical solution (28) facilitates the (also analytical) computation of matrix $\mathbf{M}_{\chi,i}$, which is finally found as

$$\mathbf{M}_{\chi,i} = \Phi_{\text{focus},i} \mathbf{T}_{\text{focus},i} \Phi_{\text{focus},i}^T. \quad (32)$$

Here, the matrix $\Phi_{\text{focus},i}$ is

$$\Phi_{\text{focus},i} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -\frac{\dot{z}_{\text{TD},i}}{5} & \frac{g}{2\tilde{p}_{z,i}} & 2T_{s,i} & 0 & 0 \\ \frac{\tilde{p}_{z,i}}{3\tilde{p}_{z,i}} \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} & \frac{5T_{s,i}}{3} & -\frac{2}{3} \end{bmatrix}. \quad (33)$$

With $v_1 = (z_{\text{floor},i} - z_{\text{TD},i} - \dot{z}_{\text{TD},i}T_{s,i})$, $v_2 = \frac{gT_{s,i}}{2} - \dot{z}_{\text{TD},i}$, and $v_3 = 2\tilde{p}_{z,i}T_{s,i}^2 + \frac{g}{2}$. All time integrals are combined in

$$\mathbf{T}_{\text{focus},i} = \begin{bmatrix} \frac{T_{s,i}^2}{12} & \frac{T_{s,i}^3}{4T_{s,i}} & \frac{3T_{s,i}^4}{40} & \frac{T_{s,i}^5}{8T_{s,i}} & \frac{5T_{s,i}^6}{84} \\ \frac{T_{s,i}^3}{12} & \frac{4T_{s,i}^4}{45} & \frac{T_{s,i}^5}{9T_{s,i}} & \frac{8T_{s,i}^6}{3T_{s,i}} & \frac{5T_{s,i}^7}{5T_{s,i}} \\ \frac{3T_{s,i}^4}{40} & \frac{T_{s,i}^5}{8T_{s,i}} & \frac{9T_{s,i}^6}{3T_{s,i}} & \frac{3T_{s,i}^7}{16T_{s,i}} & \frac{5T_{s,i}^8}{T_{s,i}} \\ \frac{40}{15} & \frac{12}{105} & \frac{112}{40} & \frac{40}{225} & \frac{72}{15} \\ \frac{T_{s,i}^5}{5T_{s,i}} & \frac{8T_{s,i}^6}{5T_{s,i}} & \frac{3T_{s,i}^7}{5T_{s,i}} & \frac{16T_{s,i}^8}{T_{s,i}} & \frac{T_{s,i}^9}{25T_{s,i}^{10}} \\ \frac{15}{84} & \frac{105}{72} & \frac{40}{72} & \frac{225}{15} & \frac{15}{396} \end{bmatrix}. \quad (34)$$

Now, with the matrix $\mathbf{M}_{\chi,i}$ and differentiating (31) with respect to $\tilde{p}_{\chi,i}$, we find the optimal parameter

$$\tilde{p}_{\chi,i}^* = -\frac{\mathbf{r}_{\chi,i}^T \mathbf{M}_{\chi,i} \mathbf{p}_{\chi,i,0}}{\mathbf{r}_{\chi,i}^T \mathbf{M}_{\chi,i} \mathbf{r}_{\chi,i}}, \quad (35)$$

which minimizes the mean square deviation as defined above. With (35), (20) turns into

$$\mathbf{p}_{\chi,i} = \underbrace{\left(\mathbf{I} - \frac{\mathbf{r}_{\chi,i} \mathbf{r}_{\chi,i}^T \mathbf{M}_{\chi,i}}{\mathbf{r}_{\chi,i}^T \mathbf{M}_{\chi,i} \mathbf{r}_{\chi,i}} \right)}_{\Omega_{\chi,i}} \underbrace{\mathbf{B}_{\chi,i}^T (\mathbf{B}_{\chi,i} \mathbf{B}_{\chi,i}^T)^{-1}}_{\mathbf{B}_{\chi,i}^+} \mathbf{b}_{\chi,i}, \quad (36)$$

which directly maps the horizontal boundary conditions $\mathbf{b}_{\chi,i}$ to the appropriate polynomial parameter vectors $\mathbf{p}_{\chi,i}$ (including best force focus). If horizontal CoM TD target positions (or similarly: takeoff velocities) are used as boundary conditions, as in [1], (36) provides the solution to the problem.

C. Foot Step Targeting and Leg Crossover Avoidance

In [1], foot positions could not be controlled directly, which caused problems with leg crossover (see Fig. 7). In addition, when precise foot placement is required—when running over stepping stones, as in Fig. 1—the method failed to provide any guarantee of safe stepping. To address these drawbacks, in this paper, we aim at an explicit solution for foot-step targeting. Setting $\bar{\chi}_{\text{int},i} = \chi_{\text{foot},i}$ in (27), and inserting (36), we can solve for the desired upcoming CoM TD position¹ $\chi_{\text{TD},i+1,\text{des}}$, which corresponds to the desired foot location $\chi_{\text{foot},i}$. Resubstituting this particular $\chi_{\text{TD},i+1,\text{des}}$ in (36) yields

$$\mathbf{p}_{\chi,i} = \underbrace{[(\mathbf{I} - \mathbf{e}_{\chi,i}^\oplus \mathbf{e}_{\chi,i}^T) \Omega_{\chi,i} \Pi_{\chi,i}]}_{\mathbf{A}_{\text{TD},\chi,i}} \mathbf{e}_{\chi,i}^\oplus \begin{bmatrix} \chi_{\text{TD},i} \\ \dot{\chi}_{\text{TD},i} \\ \chi_{\text{foot},i} \end{bmatrix}. \quad (37)$$

Here, $\mathbf{A}_{\text{TD},\chi,i}$ maps the i th TD state to $\mathbf{p}_{\chi,i}$ and the specific pseudoinverse $\mathbf{e}_{\chi,i}^\oplus = \frac{\Omega_{\chi,i} \pi_{\chi,i}}{\mathbf{e}_{\chi,i}^T \Omega_{\chi,i} \pi_{\chi,i}}$ of $\mathbf{e}_{\chi,i}$ maps the i th foot position. The matrix $\Pi_{\chi,i}$ combines the first two column vectors of $\mathbf{B}_{\chi,i}^+$, while $\pi_{\chi,i}$ is its final column. Note that the third and fourth boundary conditions in (18) are implicitly accounted for. We will now use all previewed desired footholds $\chi_{\text{foot},i} = \chi_{\text{foot},\text{des},i}$ (except the first one) and the final takeoff velocity $\dot{\chi}_{\text{TO},N} = \mathbf{0}$ as constraints and solve for the first foothold $\chi_{\text{foot},1}$ (control variable) and all future horizontal polynomial parameter vectors $\mathbf{p}_{\chi,i}$, which yield perfect tracking of the future desired footholds. By combining the TD state into $\tau_{\chi,i} = [\chi_{\text{TD},i}, \dot{\chi}_{\text{TD},i}]^T$, (37) becomes

$$\mathbf{p}_{\chi,i} = \mathbf{A}_{\text{TD},\chi,i} \tau_{\chi,i} + \mathbf{e}_i^\oplus \chi_{\text{foot},i}. \quad (38)$$

Each upcoming TD state can be previewed as

$$\tau_{\chi,i+1} = \begin{bmatrix} \chi_{\text{TD},i+1} \\ \dot{\chi}_{\text{TD},i+1} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{t}_{\chi,i}^T(T_{s,i}) + T_{f,i} & \mathbf{t}_{\chi,i}^T(T_{s,i}) \\ \mathbf{t}_{\chi,i}^T(T_{s,i}) \end{bmatrix}}_{\mathbf{S}_i} \mathbf{p}_{\chi,i}. \quad (39)$$

Starting from $i = 1$ and propagating (38) and (39) forward, we find the following expression for the polynomial parameter vector of the N th (i.e., final) previewed stance phase:

$$\mathbf{p}_{\chi,N} = \mathbf{G}_{\chi,1} \mathbf{A}_{\text{TD},\chi,1} \tau_{\chi,1} + \sum_{i=1..N} (\mathbf{G}_{\chi,i} \mathbf{e}_i^\oplus \chi_{\text{foot},i}). \quad (40)$$

The matrices $\mathbf{G}_{\chi,i} = \mathbf{G}_{\chi,i+1} \mathbf{A}_{\text{TD},\chi,i+1} \mathbf{S}_i$ are evaluated by starting with $\mathbf{G}_{\chi,N} = \mathbf{I}_{6 \times 6}$ and iterating backwards until $i = 1$. Now, with (2) and (40), we find the horizontal takeoff velocity after the final previewed stance phase as

$$\dot{\chi}_{\text{TO},N} = \mathbf{t}_{\chi}^T(T_{s,N}) \mathbf{p}_{\chi,N}. \quad (41)$$

¹This is why $\chi_{\text{TD},i+1,\text{des}}$ was called an “intermediate control target” earlier.

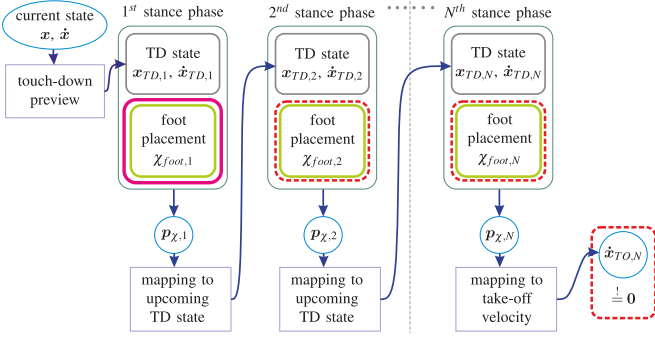


Fig. 5. CoM TD adjustment for foot targeting.

To guarantee stability, we choose $\dot{x}_{TO,N} = 0$ as terminal constraint, i.e., the controller presumes the CoM to come to a full stop after the final previewed contact phase.

Note that in the same way for the computation of the final vertical polynomial parameter vector, we use the terminal constraint $\dot{z}_{TO,N} = 0$ instead of a desired apex height.

As mentioned above and shown in Fig. 5, we aim to achieve the terminal constraint and all desired foot targets $\chi_{foot,des,i}$ other than the first one (which is sacrificed in order to serve as a control variable). We, therefore, solve (40) and (41) for $\chi_{foot,1}$ that finally yields the sought-after first foot placement

$$\chi_{foot,1} = \frac{-t_{\chi}^T(T_{s,N}) (G_{\chi,1} A_{TD,\chi,1} \tau_{\chi,1} + \eta)}{t_{\chi}^T(T_{s,N}) G_{\chi,1} e_{\chi,1}^{\oplus}} \quad (42)$$

$$\eta = \sum_{i=2..N} (G_{\chi,i} e_{\chi,i}^{\oplus} \chi_{foot,des,i}).$$

Now, we solve for all horizontal polynomial parameter vectors $p_{\chi,i}$ by alternately evaluating (38) and (39). As foot positions in (38), we use $\chi_{foot,i} \in \{\chi_{foot,1}, \chi_{foot,des,2}, \dots, \chi_{foot,des,N}\}$. During stance, we freeze the first foot position $\chi_{foot,1}$ and polynomial parameter vector $p_{\chi,1}$ (feedforward) and use the second foothold $\chi_{foot,2}$ as the control variable instead. That way, even in the face of unknown perturbations, the foot targets are continuously adjusted. Equation (42) is adopted accordingly.

One feature of our framework is that due to the multistep preplanning, both future foot aim points $\chi_{foot,1}$ and $\chi_{foot,2}$ (i.e., the aim points of the left and right foot) are known at all times, which facilitates foot trajectory generation. In this study, we implemented the foot trajectories as polynomials. The achieved precise foot targeting is particularly interesting for running over 3-D stepping stones or other restricted surfaces.

An additional feature of precise foothold targeting is that leg crossover can be explicitly avoided. This is especially helpful for running in sharp turns (see Fig. 7). Therefore, the originally preplanned footholds can be adjusted such that the left foot always passes by the right foot on the left, and vice versa. At the same time, the Euclidean distance of the adjusted footholds from the originally planned ones should be minimal. This way, the legs can be prevented from twisting around each other. To achieve this goal, we use an adjustment heuristic, as shown in Fig. 7. In the shown example, we preview four foot positions,

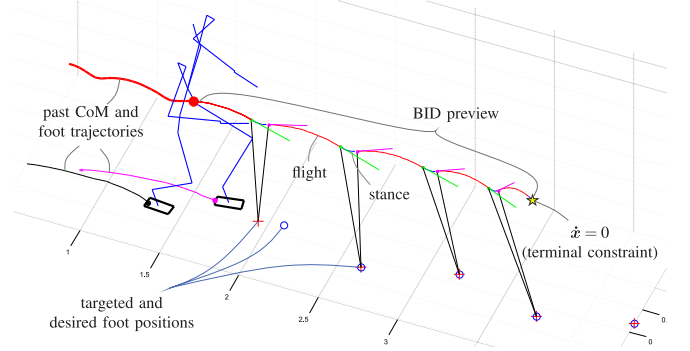


Fig. 6. BID preview of Toro (displayed as stickman) running in OpenHRP. All desired foot positions (except the first one) are previewed to be perfectly tracked.

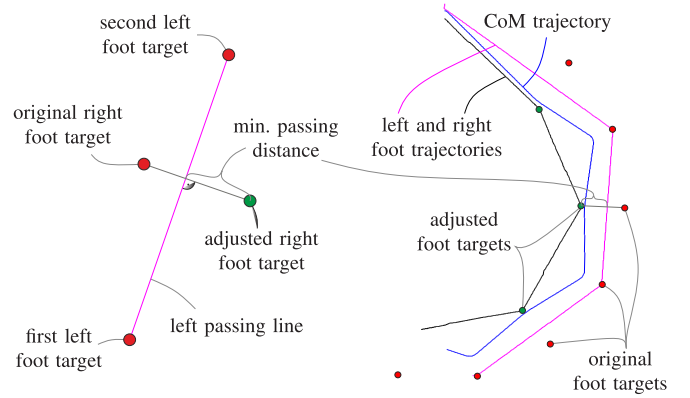


Fig. 7. Leg crossover avoidance. (Left) Scheme (depicted for left pass). (Right) Simulation output.

i.e., two for each foot. The method adjusts the second/third desired footstep (i.e., the projection shown in Fig. 7, left, is applied twice), such that the swing feet can safely swing from the first/second foothold to the third/fourth one. The fourth foothold remains unchanged to achieve good long-term tracking of the original desired foot locations.

VI. STATE FEEDBACK CONTROL

In the nominal case (no perturbations), the force profiles and foot aim points as derived in the previous sections assure that for any initial conditions after the first stance phase, all desired boundary conditions from Sections V-A and V-B are fulfilled (deadbeat control). Therefore, planning once per step or even preplanning a whole sequence of upcoming steps would be sufficient. Yet, to cope with perturbations, we propose a state feedback control method, which is based on continuous replanning of the desired contact forces throughout both flight and stance phases. To this end, during flight, the previewed CoM TD state is updated (see Fig. 8) via (7). In contrast with [1] (no feedback during stance), during stance, the first takeoff state is predicted

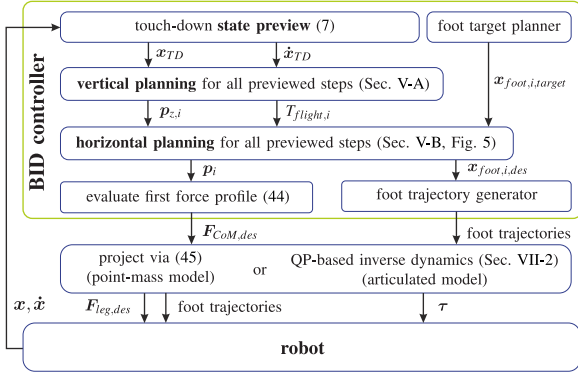


Fig. 8. Computation flow of BID controller (outline).

via

$$\begin{bmatrix} \sigma_{TO,1} \\ \dot{\sigma}_{TO,1} \end{bmatrix} = \underbrace{\begin{bmatrix} \sigma \\ \dot{\sigma} \end{bmatrix}}_{\text{feedback}} + \underbrace{\begin{bmatrix} \mathbf{t}_{\sigma}^T(T_{s,1}) - \mathbf{t}_{\sigma}^T(t_s) \\ \mathbf{t}_{\sigma}^T(T_{s,1}) - \mathbf{t}_{\sigma}^T(t_s) \end{bmatrix} \mathbf{p}_{\sigma,1}}_{\text{preview}}, \quad \sigma \in \{x, y, z\}. \quad (43)$$

Here, $\mathbf{t}_{\sigma}^T(t)$ and $\mathbf{t}_{\sigma}^T(t)$ are the time-mapping row vectors from (2). They are evaluated for the first total stance time $T_{s,1}$ and the current time in stance $t_s \in [0, T_{s,1}]$ to predict how much of an offset is expected if for the remaining time in step the current force profile (encoded by $\mathbf{p}_{\sigma,1}$) is applied. This offset is added to the current measured state to predict the takeoff state, which in turn is used to compute the upcoming CoM TD state. Note that after TD, the force profile of the current stance phase is frozen and commanded to the robot as feedforward. The main advantage of our state feedback during stance is that the foot aim points are continuously updated to avoid discontinuities in the foot reference trajectories.

Note that during flight, the first upcoming foot position is one of the main control inputs [see (42)]. While all other future footsteps are previewed to coincide with the desired foot target locations (see Fig. 6), the nominal position of the first foot is an output of the controller. Depending on the limitations at hand (e.g., limited allowable supporting area), this nominal foot aim point may have to be projected to a feasible one, resulting in deviations from the nominal deadbeat behavior.

VII. GUARANTEEING LEG FORCE FEASIBILITY

The desired 3-D force acting on the CoM can be computed for a given time in stance t_s as

$$\mathbf{F}_{\text{CoM,des}}(t_s) = m \begin{bmatrix} \mathbf{t}_{\ddot{x}}^T(t_s) \mathbf{p}_{x,1} \\ \mathbf{t}_{\ddot{y}}^T(t_s) \mathbf{p}_{y,1} \\ \mathbf{t}_{\ddot{z}}^T(t_s) \mathbf{p}_{z,1} \end{bmatrix} \quad (44)$$

i.e., the polynomial of the first force profile is evaluated. The corresponding desired leg force $\mathbf{F}_{\text{leg,des}}$ is found by reordering (1). The polynomial parameters were chosen to result in the best achievable focus of the leg forces with the ground. Yet, for physical robots, feasibility is not guaranteed.

A. Point-Mass Point-Feet Model

One obvious example is when the robot is modeled as point-mass with point feet. In that case, the leg force is constrained to point along the unit vector $\mathbf{u}_{x,f}$, pointing from CoM to point foot. As the other two spatial directions are unactuated, the desired leg force $\mathbf{F}_{\text{leg,des}}$ has to be projected to the feasible direction² as

$$\mathbf{F}_{\text{leg},f} = \mathbf{u}_{x,f} \mathbf{u}_{x,f}^T \mathbf{F}_{\text{leg,des}}. \quad (45)$$

Assuming sufficient ground friction, $\mathbf{F}_{\text{leg},f}$ can be safely commanded to the point-mass point-foot model.

B. Articulated Multibody Model

As in our previous work on walking [11], the main idea of our BID control concept is to first focus on the robot's CoM dynamics and the problem of foot placement, which in our view are the key challenges of locomotion. A point-mass model can be sufficient to address these issues. Once CoM dynamics and foot placement are solved, they need to be embedded into a more general control framework to make them available for articulated multibody models such as simulated or real humanoid robots. To this end, we use an inverse dynamics-based whole-body control framework similar to [24] and [27]. It solves a single QP that tries to satisfy the specified tasks as best as possible while guaranteeing feasibility. The tasks include foot trajectory tracking, upper-body posture control, overall joint posture control, and a centroidal momentum task [28], which can be subdivided into linear and angular momentum tasks. Most of the tasks (excluding the BID controller) include a task space PD control component. The desired linear force on the CoM from the BID controller (44) is directly fed into our linear momentum task. The angular momentum task aims to regulate the robot's overall angular momentum to zero. The foot trajectories from the BID controller form the direct input to the whole-body foot task (see Fig. 8). Note that feasibility here relates only to ground reaction wrenches and joint torques, while stability or balance (depending on the physical limitations of the robot at hand) is not guaranteed.

VIII. ENHANCING KINEMATIC FEASIBILITY

A major issue concerning the porting of BID control to kinematically restricted robot models (such as humanoid robots) is that the BID controller does not naturally consider any kinematic limitations. In the case of high desired velocities and accelerations or strong perturbations, BID may result in unrealistic high required leg lengths. In the next two subsections, we will present two methods to ease this problem (while not guaranteeing feasibility): one for finding nominally feasible gait parameters and one for online touch-down leg length adjustment. Both methods assume that the distance from CoM to foot in the BID preview correlates to the leg length in a multibody model. This is an approximation, of course.

²Note that for more complex robots, this projection may not be necessary.

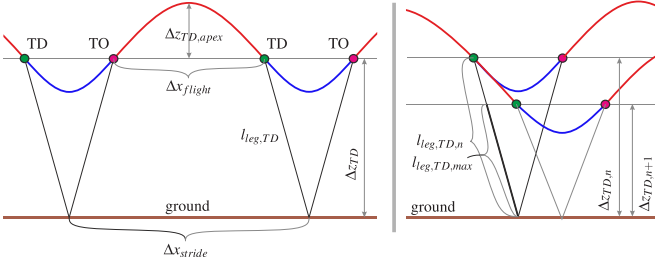


Fig. 9. Correlations for stationary running.

A. Nominally Feasible Gait Design

For a periodic running gait (assumed here), the CoM height at TD z_{TD} equals the one at takeoff z_{TO} (see Fig. 9). With energy conservation ($\dot{z}_{TO}^2 = 2g\Delta z_{TD,apex}$) and with (5), we can derive the time of flight T_f (i.e., from TO to TD) as

$$T_f = \sqrt{\frac{8\Delta z_{TD,apex}}{g}}. \quad (46)$$

Here, $\Delta z_{TD,apex}$ denotes the height difference between apex and TD. For a desired flight percentage $f_{flight} = \frac{T_f}{T_s + T_f}$ and with the mean horizontal speed (e.g., derived from a joystick input) $v_{mean} = \frac{\Delta x_{stride}}{T_s + T_f}$, we get

$$T_f = f_{flight} \underbrace{(T_s + T_f)}_{\frac{\Delta x_{stride}}{v_{mean}}} \leq f_{flight} \frac{\Delta x_{stride,max}}{v_{mean}}. \quad (47)$$

Here, the inequality indicates that the time of flight should be small enough, such that a maximum desirable stride length $\Delta x_{stride,max}$ is not exceeded. By combining (46) and (47), we find a condition for the maximum allowable height difference between apex and TD

$$\Delta z_{TD,apex} \leq \underbrace{\frac{g}{8} \left(f_{flight} \frac{\Delta x_{stride,max}}{v_{mean}} \right)^2}_{\Delta z_{TD,apex,max}}. \quad (48)$$

A second condition for nominal kinematic feasibility is that a maximum allowable TD leg length $l_{leg,TD,max}$ is not exceeded. By inspection of Fig. 9(left), we find the following condition for the CoM TD height:

$$\Delta z_{TD} \leq \underbrace{\sqrt{l_{leg,TD,max}^2 - \left(\frac{\Delta \hat{x}_{stride} - \Delta \hat{x}_{flight}}{2} \right)^2}}_{\Delta z_{TD,max}}. \quad (49)$$

Here, $\Delta \hat{x}_{stride} = v_{mean} (T_s + T_f)$ and $\Delta \hat{x}_{flight} = v_{mean} T_f$ denote the approximated (assuming constant horizontal velocity) distances traveled during a whole stride and during a single flight phase, respectively. With the described adjustments of apex (48) and TD height difference (49), the nominal desired TD and apex

height difference (above each upcoming floor height) become

$$\begin{aligned} \Delta z_{TD,des} &= \min(\Delta z_{TD,nominal}, \Delta z_{TD,max}) \text{ and} \\ \Delta z_{apex,des} &= \Delta z_{TD,des} + \min(\Delta z_{TD,apex,nominal}, \Delta z_{TD,apex,max}) \end{aligned} \quad (50)$$

where $\Delta z_{TD,nominal}$ and $\Delta z_{TD,apex,nominal}$ act as upper limits. In addition, the nominal stance time can be computed with (46) as

$$T_s = \frac{1 - f_{flight}}{f_{flight}} \sqrt{\frac{8\Delta z_{TD,apex}}{g}}. \quad (51)$$

That way, the design parameters introduced in Section III (i.e., $\Delta z_{TD,des}$, $\Delta z_{apex,des}$, and T_s) are deduced from desired flight percentage f_{flight} , maximum desired stride length $\Delta x_{stride,max}$, maximum desired TD leg length $l_{leg,TD,max}$, and mean horizontal speed v_{mean} to maximize kinematic feasibility.

B. Active Leg Length Control

In the case of strong perturbations, the leg lengths resulting from the BID controller (and the applied foot trajectory generator) may not comply with kinematic limitations of multibody robots. To ease this problem, we adjust the original BID plan via the following iteration scheme [see Fig. 9(right)]

$$\Delta z_{TD,i,n+1} = \min\left(\Delta z_{TD,des}, \frac{l_{leg,TD,max}}{l_{leg,TD,i,n}} \Delta z_{TD,i,n}\right). \quad (52)$$

The TD height difference is iteratively adjusted (if it does not exceed the nominal TD height $\Delta z_{TD,des}$) such that for each stance phase i , the resulting TD leg length $l_{leg,TD,i}$ does not exceed the maximum desired TD leg length $l_{leg,TD,max}$ (similar to the rest length of SLIP models). Here, n denotes the iteration count. For each iteration, the complete BID preview has to be reevaluated.

IX. SIMULATIONS AND EVALUATION OF BIOLOGICALLY INSPIRED DEADBEAT CONTROL

A. Biologically-Inspired-Deadbeat-Based Point-Mass Simulations

To test the performance and robustness of the proposed control framework, we performed numerous simulations. For the first set of simulations, we considered a point-mass with two massless point-feet. Fig. 10 shows the results of a robustness examination for three different constant external forces. From top to bottom, the figure shows phase plots for three simulations. Each simulation was set up in the following way: no perturbation during the first 4 s, then 4 s of constant force acting (magnitude: -50 N [corresponding to $\approx 10\%$ of the robot's mass (here, 50 kg)], force direction: purely x , y , and z , respectively), followed by 4 s of no perturbation. Here, $\Delta x = x - x_{joystick}$ and $\Delta y = y - y_{joystick}$ denote the errors w.r.t. the nominal horizontal CoM position $\mathbf{x}_{joystick} = [x_{joystick}, y_{joystick}]^T$, which was computed from a virtual joystick input. The stars denote the initial states. The phase plots show that for perturbed and unperturbed phases, the system very quickly converges to corresponding limit cycles. Note that the perturbation forces in the shown

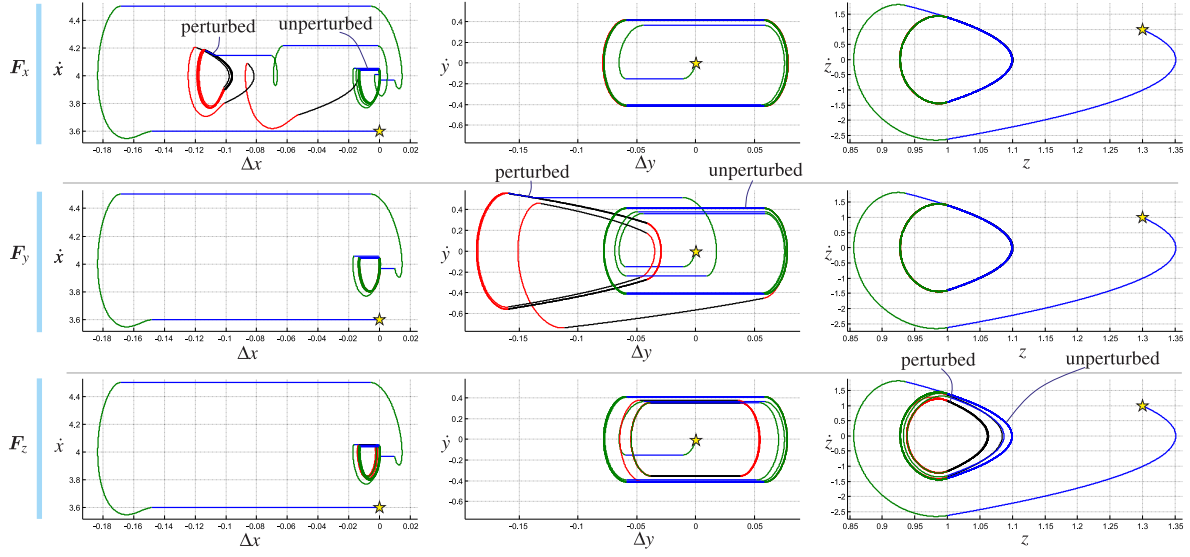


Fig. 10. Robustness examination (point-mass) for different constant external forces. Perturbation inactive: stance green, flight blue. Perturbation active: stance red, flight blue.

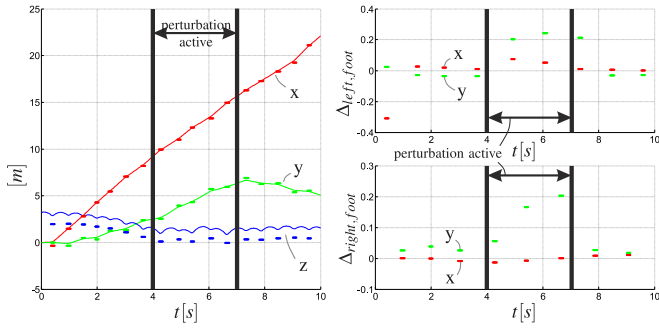


Fig. 11. CoM trajectories and foot target tracking performance in nominal and perturbed case (point-mass simulation over stepping stones). Perturbing force: 30 N in the y -direction.

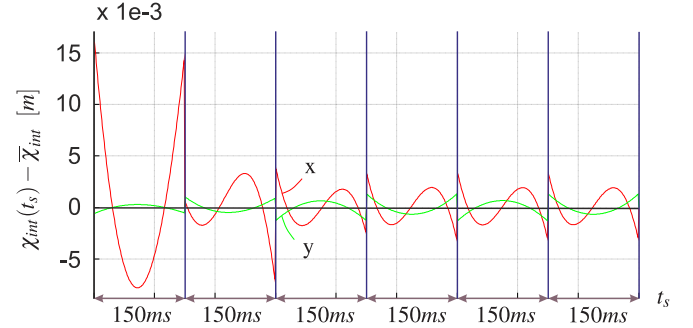


Fig. 12. Deviation of force intersection point $\chi_{\text{int}}(t_s)$ from mean intersection point $\bar{\chi}_{\text{int}}$ in case that (in contrast with other point-mass simulations) the forces are not projected. Along the time axis, the stance phases are pieced together.

simulations were kept comparably low to increase readability of the plots.

We performed many further BID-based simulations with a bipedal point-mass robot, which showed a very high robustness of the basic BID controller. For extreme perturbations, the leg length could grow to unrealistic levels (due to the constant TD height). To assure leg length feasibility, the method from Section VIII-B can be applied. The controller is most sensitive against strong unknown perturbations that point toward the ground. Here, the maximum permanent force the controller could withstand in simulation was -750 N, i.e., 1.5 times the robot's weight. For higher forces, the robot's CoM would hit the ground.

Fig. 11 shows the result of a simulation in which the point-mass robot was running over three-dimensional stepping stones (see also Fig. 1). The left subplot shows the robot's foot positions (bars, only active during stance) and CoM positions (continuous curves). The right subplots show the difference between desired and achieved foot positions. Nominally, the foot target positions are tracked well (not perfectly since projection (45) was active),

whereas in the case of perturbations, they deviate. This is necessary to stabilize the CoM motion against the perturbation. After the perturbation is removed, good tracking is regained after a single step.

Fig. 12 shows how far the force intersection point $\chi_{\text{int}}(t_s)$ deviates from the mean intersection point $\bar{\chi}_{\text{int}}$ (i.e. the stance foot position) for the case that the *desired force profiles* are not projected. In the shown simulation, the robot starts at zero speed and then runs at 2 m/s. The stance time is set to 150 ms. The initial range of deviation is about 22 mm, while for stationary running it is about 6 mm. This shows that the original (non-projecting) method is well applicable for smallfooted robots and that (45) typically has minor influence.

B. Quadratic-Program-Based Multibody Simulations

To prove the applicability of the BID control framework, we embedded it into the QP-based whole-body controller from Section VII-B and performed full-body simulations of the humanoid robot Toro [29] in OpenHRP [30]. It has to be noted

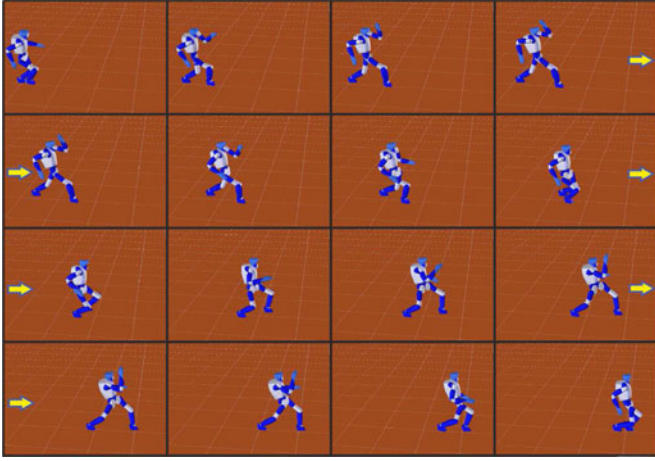


Fig. 13. Toro [29] running in OpenHRP [30] at 5 m/s.

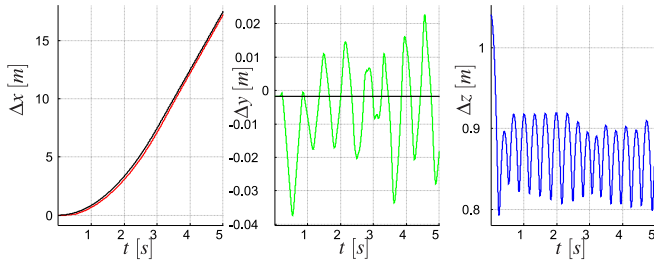


Fig. 14. Toro's CoM while running in OpenHRP at 0–5 m/s.

that Toro's joint torque and velocity limits were omitted in the simulations. Fig. 13 shows Toro running at 5 m/s. The corresponding CoM motion (colored) as compared to its reference (black) is depicted in Fig. 14. The gait parameters $\Delta z_{TD,des}$, $\Delta z_{apex,des}$, and T_s were computed via the method from Section VIII-A to make such a high running speed kinematically feasible for Toro. Following intuitive design parameters (as described in Section VIII-A) were chosen: desired flight percentage $f_{flight} = 0.7$, maximum desired stride length $\Delta x_{stride,max} = 1.4$ m, maximum desired TD leg length $l_{leg,TD,max} = 0.86$ m, nominal TD height $\Delta z_{TD,nominal} = 0.86$ m, and nominal height difference between TD and apex $\Delta z_{TD,apex,nominal} = 0.06$ m. The target velocity (derived from a virtual joystick input and used as v_{mean}) ramped up from 0 to 5 m/s until second 3 and then stayed constant. Two important human-like features evolved: first, natural arm swing motions (see also multimedia attachment) that facilitate the angular momentum regulation and contribute to the CoM manipulation and, second, stretched hind legs at the end of stance. This shows that the combination of BID and whole-body control can automatically create human-like motions such that the effect of the various cost functions and their weights can be examined.

An important quality of a running controller is its reactivity and robustness. Without that quality, the OpenHRP running simulations would fail due to the overdeterminedness of tasks (such as CoM force and angular momentum control, posture control, etc.), tracking errors and energy losses at impact. To investigate this quality of our combined BID and whole-body

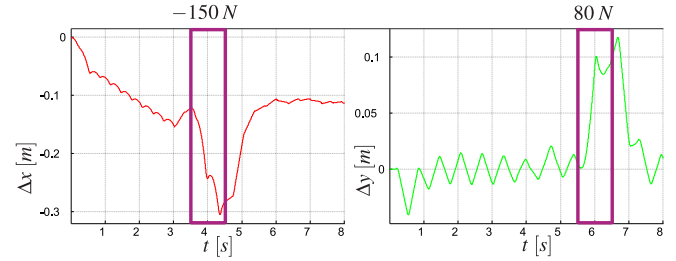


Fig. 15. CoM error during push-recovery simulation.

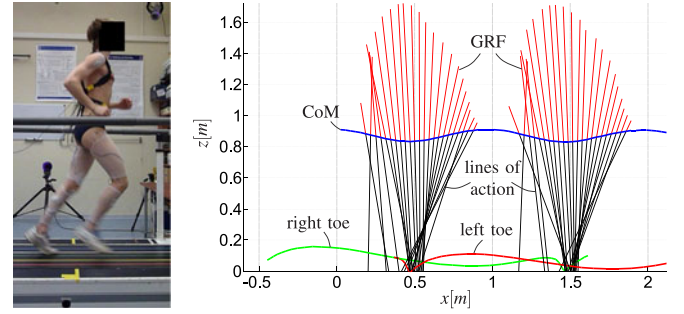


Fig. 16. Human-running experiment. (Left) Subject running on force plate treadmill. (Right) Trajectories and GRF.

control framework, we performed multiple simulations in which the robot was subject to external perturbations. One of these simulations is shown in Fig. 15. It displays the errors in horizontal CoM position with regard to the joystick reference. Toro runs at 3 m/s (after ramping up from 0 m/s until second 3). From second 3.5 to 4.5, it is subject to a backwards pointing external force of -150 N and between second 5.5 and 6.5 to a lateral force of 80 N (both constant and unknown). The controller compensates for these perturbations and recovers after just a few steps. The steady state error of about 0.1 m in the x -direction can be explained by the fact that the foot step (not the CoM) is planned to coincide with the joystick reference (aside from a sideward offset) at the instant of TD, while the continuous joystick reference keeps moving throughout stance. The kinematic feasibility of the running gait under these strong perturbations was facilitated by the methods from Section VIII.

The OpenHRP simulations of Toro running show our control framework's robustness and reliability. It is thus a promising concept for a future, more detailed comparison between human and humanoid running and prediction of human behavior.

X. COMPARISON WITH HUMAN EXPERIMENT

The BID controller had been inspired by observations from human-running experiments. In the previous section, we showed its high robustness, which substantiates its applicability for humanoid running control. Now, the question is how well do the BID control outputs fit to the ones observed in human-running experiments? Thus, we close the loop by comparing the corresponding forces and CoM trajectories. Fig. 16 (left) shows a human subject running on a force plate treadmill, its posture being tracked via markers. On the right side of the figure, the corresponding CoM and toe trajectories are shown. It becomes

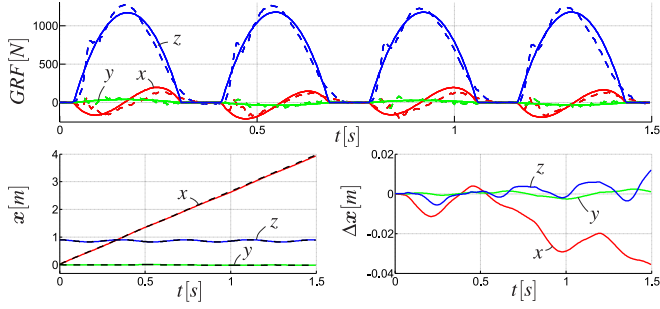


Fig. 17. Comparison of human experiment data [31], [32] (dashed) and output of matched BID simulation. top: GRF. (Left) CoM position. (Right) CoM error.

apparent that the lines of action of the GRF in humans are not as strictly focused as the ones designed in our BID controller (compare to Figs. 3 and 4). This shows that humans make use of angular momentum during running, while the CoP remains in the ball of the foot (compare toe trajectories). Fig. 17 shows the corresponding force profiles and CoM trajectories and overlays them to a “matched”³ BID simulation. The force profiles match quite well. The main differences are the initial impacts, slightly higher vertical force maximum, and lower final force slope in humans as compared with the BID simulation. The CoM positions are very consistent [errors in the range of several millimeters (x -direction drifting due to slight timing mismatch)]. From these observations, we infer that BID control sufficiently approximates the GRF in human running to allow for decent insights into human-running control. Yet, not surprisingly, the observed differences motivate further examination of human-running control.

XI. DISCUSSION AND OUTLOOK

A. Strengths and Limitations of Current Control Framework

In this paper, we provided a closed-form solution to 3-D running. The proposed control framework yields leg force profiles that are independent of the specific hardware design of a particular robot, i.e., it is generic. In addition, it might be used to identify required actuator characteristics for new robots.

For our simulations, we used a standard PC (3.3 GHz, quad-core, Win7 64 bit). In our MATLAB/Simulink simulation setup and using 1 ms as sampling time, we were able to execute all BID control-related computations in real time.

The force profiles as derived in Sections V-A and V-B nominally lead to perfect tracking after just one stance phase (dead-beat control), i.e., the controller is perfectly stable. In the case of actuation limits, the control commands may have to be adjusted (e.g., via (45) for point-mass point-feet robots); therefore, stability cannot be guaranteed. Yet, our simulations show the high robustness of the controller, even in case of constraints.

In our control framework, impact-free state transitions are assumed (compare Fig. 2). The impact losses in real systems

³By “matching,” we mean that basic gait parameters, such as stance time, apex, and TD height, and also the human subject’s and the model’s mass have to be aligned manually to best fit with the observed data. Otherwise, a comparison—especially in time domain—would be impossible/useless.

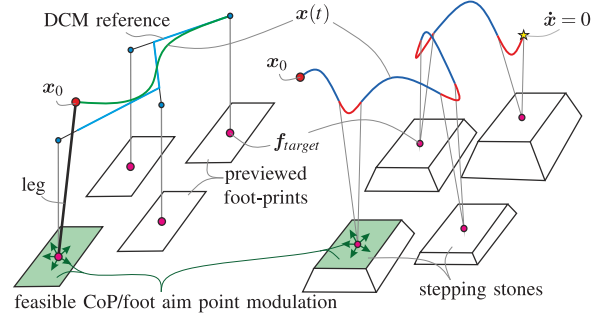


Fig. 18. Analogy of DCM (walking) and BID (running); stepping stones are comparable with finite-sized feet.

will cause perturbations. In any case, due to its high robustness in simulations, we expect good performance of the controller.

A drawback of our current control setup when compared with human running is the missing toe-off motion. In the current setup, the feet are aligned with the ground during contact. Toe-off motion (especially during single support) is usually classified as a challenging task. Consequently, it has to be tackled in future research to enhance the capabilities of humanoid running and make it more comparable with its natural counterpart.

B. Comparison With Other Works

When compared with SLIP control, the main feature of our presented BID controller is its analyticity, which allows for explicit solutions, e.g., for 3-D CoM trajectories and foot-step placement during running. Some features of the impressive work of Raibert [13], such as apex height control and forward speed control via foot placement, show major similarities to our work. Yet, BID control provides analytic solutions for planning and control as compared to Raibert’s three-parted and rather heuristic running controller.

Although the method proposed in [11] (based on the divergent component of motion (DCM), a.k.a. capture point) handles a different form of locomotion, namely walking, on closer inspection, its overall control framework shows similarities with the BID controller proposed in this paper. The first analogy is the preview of several (typically three or more) future footsteps and the derivation of feasible force profiles that nominally track them. The second analogy is related to the modulation and potential projection of the desired forces such that they comply with the contact constraints. In the case of DCM control, this modulation/projection consists of leg force modulation and projection of a desired center of pressure (CoP) to the feasible foot supporting area, respectively. The proposed BID controller, in comparison, modulates the first upcoming stance foot position and all previewed leg force profiles, while projecting the foot position to a feasible one in the case of limited allowable contact area (see Fig. 18).

C. Potential Usage, Extensions, and Future Work

One interesting aspect in human running is the CoP motion from heel to toe (as observed, for example, during medium speed jogging). This effect can be observed in Fig. 16 (intersection of the black force lines with the ground). This means that, while

in humanoid locomotion, one usually keeps the nominal CoP as close to the foot center as possible (as we did in this paper as well; see Fig. 4) to increase the likelihood of feasible leg forces; it can be more optimal to actually move the CoP from heel to toe during stance. A simple trick to produce such nominal CoP motion using our proposed BID control framework would be to set the virtual foot positions below the actual ground. That way, the intersection points of the force lines with the actual ground (which correspond to the CoP) would show a heel-to-toe motion. How to make use of such virtual foot point adjustment and the correlating heel-to-toe motion is a question that we will examine in our future research.

In this study, we work with locally flat stepping stones (see Fig. 1). However, we suppose that arbitrary ground surfaces could be handled, the major difficulty being to incorporate the ground profile in the search for the first (i.e., actively adjusted) footstep. That way, 3-D foot locations on arbitrary known terrain could be targeted. Naturally, robust foot trajectories for blind running are another interesting research topic. The BID algorithm may also be applied to problems such as hopping and jumping. We also expect that quadrupedal gaits, such as bounding/galloping and trotting, can be achieved.

With regard to motion science, starting from the presented work, we plan to implement tools for human/humanoid running comparison and to perform cooperative human-running experiments specifically designed for that purpose.

XII. CONCLUSION

In this paper, we describe the BID controller, a concept for 3-D bipedal running. It encodes the leg forces during stance as polynomials. The controller facilitates agile, precise, and versatile running motions. It has deadbeat properties and is very robust against external perturbations. The method facilitates explicit foot targeting, running over 3-D stepping stones and leg crossover avoidance, and kinematic feasibility enhancement. We embedded the BID controller into a QP-based whole-body controller (similar to [24]) to achieve running with the humanoid Toro [29] in simulation. We achieved running speeds of up to 5 m/s and demonstrated push recovery. The resulting CoM trajectories and GRF were compared with human-running data and showed decent consistency. The combination of BID control—tackling the problem of CoM manipulation and balance, i.e., based on a highly reduced model—and QP-based whole-body control shows promising results and is expected to provide new insights into human(oid) movement and control.

ACKNOWLEDGEMENTS

The authors would like to thank C. Rode, R. Müller, and M. Maus from Locomotion Laboratory (TU Darmstadt) and Imperial College London for providing the human-running data and the photo used in this study. They would also like to thank D. Clever from the Optimization in Robotics and Biomechanics Group (University of Heidelberg) for providing human experiment data and for the good discussions. They especially thank A. Mercader who significantly contributed to finding the purely analytical solution.

REFERENCES

- [1] J. Engelsberger, P. Kozłowski, and C. Ott, “Biologically inspired dead-beat controller for bipedal running in 3D,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 989–996.
- [2] J. Engelsberger, P. Kozłowski, and C. Ott, “Biologically inspired deadbeat control for running on 3D stepping stones,” in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2015, pp. 1067–1074.
- [3] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, “Efficient bipedal robots based on passive dynamic walkers,” *Science*, vol. 307, no. 5712, pp. 1082–1085, 2005.
- [4] M. Vukobratovic and Y. Stepanenko, “On the stability of anthropomorphic systems,” *Math. Biosci.*, vol. 15, pp. 1–37, 1972.
- [5] S. Kajita *et al.*, “Biped walking pattern generation by using preview control of zero-moment point,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2003, pp. 1620–1626.
- [6] P.-B. Wieber, “Trajectory free linear model predictive control for stable walking in the presence of strong perturbations,” in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2006, pp. 137–142.
- [7] S. Kajita *et al.*, “Biped walking stabilization based on linear inverted pendulum tracking,” in *Proc. Int. Conf. Intell. Robots Syst.*, 2010, pp. 4489–4496.
- [8] T. Takenaka, T. Matsumoto, and T. Yoshiike, “Real time motion generation and control for biped robot, 1st report: Walking gait pattern generation,” in *Proc. Int. Conf. Intell. Robots Syst.*, 2009.
- [9] T. Koolen, T. D. Boer, J. Rebula, A. Goswami, and J. E. Pratt, “Capturability-based analysis and control of legged locomotion. Part 1: Theory and application to three simple gait models,” *Int. J. Robot. Res.*, vol. 31, no. 9, pp. 1094–1113, 2012.
- [10] Y. Zhao and L. Sentis, “A three dimensional foot placement planner for locomotion in very rough terrains,” in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2012, pp. 726–733.
- [11] J. Engelsberger, C. Ott, and A. Albu-Schäffer, “Three-dimensional bipedal walking control based on divergent component of motion,” *IEEE Trans. Robot.*, vol. 31, no. 2, pp. 355–368, Apr. 2015.
- [12] *DARPA robotics challenge*. Accessed on: 2015. [Online]. Available: <http://www.darpa.mil/program/darpa-robotics-challenge>
- [13] M. Raibert, *Legged Robots that Balance*. Cambridge, MA, USA: MIT Press, Jan. 1985.
- [14] K. Nagasaka, Y. Kuroki, S. Suzuki, Y. Itoh, and J. Yamaguchi, “Integrated motion control for walking, jumping and running on a small bipedal entertainment robot,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2004, pp. 3189–3194.
- [15] T. Takenaka *et al.*, “Real time motion generation and control for biped robot, 4th report: Integrated balance control,” in *Proc. Int. Conf. Intell. Robots Syst.*, 2009, pp. 1601–1608.
- [16] S. Cotton, I. M. C. Olaru, M. Bellman, T. van der Ven, J. Godowski, and J. Pratt, “Fastrunner: A fast, efficient and robust bipedal robot. concept and planar simulation,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 2358–2364.
- [17] D. Lakatos, C. Rode, A. Seyfarth, and A. Albu-Schäffer, “Design and control of compliantly actuated bipedal running robots: Concepts to exploit natural system dynamics,” in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2014, pp. 930–937.
- [18] H. Geyer, A. Seyfarth, and R. Blickhan, “Compliant leg behaviour explains basic dynamics of walking and running,” *Proc. Roy. Soc. B: Biol. Sci.*, vol. 273, pp. 2861–2867, 2006.
- [19] B. Dadashzadeh, H. Vejdani, and J. Hurst, “From template to anchor: A novel control strategy for spring-mass running of bipedal robots,” in *Proc. Int. Conf. Intell. Robots Syst.*, 2014, pp. 2566–2571.
- [20] S. G. Carver, N. J. Cowan, and J. M. Guckenheimer, “Lateral stability of the spring-mass hopper suggests a two-step control strategy for running,” *Chaos*, vol. 19, no. 2, p. 026106, 2009.
- [21] H. R. Vejdani, Y. Blum, M. A. Daley, and J. W. Hurst, “Bio-inspired swing leg control for spring-mass robots running on ground with unexpected height disturbance,” *Bioinspiration Biomimetics*, vol. 8, no. 4, p. 046006, 2013.
- [22] A. Wu and H. Geyer, “The 3-D spring-mass model reveals a time-based deadbeat control for highly robust running and steering in uncertain environments,” *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1114–1124, Oct. 2013.
- [23] D. Koepl and J. Hurst, “Impulse control for planar spring-mass running,” *J. Intell. Robot. Syst.*, vol. 74, nos. 3/4, pp. 589–603, 2014.
- [24] P. M. Wensing and D. E. Orin, “High-speed humanoid running through control with a 3D-slip model,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 5134–5140.

- [25] P. Wensing and D. Orin, “3D-slip steering for high-speed humanoid turns,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 4008–4013.
- [26] H.-W. Park, S. Park, and S. Kim, “Variable-speed quadrupedal bounding using impulse planning: Untethered high-speed 3D running of MIT Cheetah 2,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 5163–5170.
- [27] T. Koolen *et al.*, “Summary of team IHMC’s virtual robotics challenge entry,” in *Proc. Int. Conf. Humanoid Robots*, 2013, pp. 307–314.
- [28] D. E. Orin, A. Goswami, and S.-H. Lee, “Centroidal dynamics of a humanoid robot,” *Auton. Robots*, vol. 35, nos. 2/3, pp. 161–176, 2013.
- [29] J. Engelsberger *et al.*, “Overview of the torque-controlled humanoid robot TORO,” in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2014, pp. 916–923.
- [30] F. Kanehiro *et al.*, “Open architecture humanoid robotics platform,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2002, vol. 1, pp. 24–30.
- [31] H.-M. Maus, S. Revzen, J. Guckenheimer, C. Ludwig, J. Reger, and A. Seyfarth, “Constructing predictive models of human running,” *Roy. Soc. Interface*, vol. 12, no. 103, 2014.
- [32] H.-M. Maus, S. Revzen, J. Guckenheimer, C. Ludwig, J. Reger, and A. Seyfarth, Data from: Constructing predictive models of human running. *Dryad Digital Repository*. 2014. [Online]. Available: <http://dx.doi.org/10.5061/dryad.r9v30>



Johannes Engelsberger received the Dipl.-Ing. degree in mechanical engineering from the Technical University of Munich, Munich, Germany, in 2009. Since then, he has been working toward the Ph.D. degree with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany.

His research interests include bipedal walking and running control, humanoid whole-body control, and mechanical design.



Paweł Kozłowski is currently working toward the master’s degree in mechatronics with the AGH University of Science and Technology, Kraków, Poland.

He worked on the development of the control algorithm for bipedal running during his internship with the Institute of Robotics and Mechatronics, German Aerospace Centre (DLR), Wessling, Germany. His research interests include mechanics and digital signal processing.



Christian Ott received the Dipl.-Ing. degree in mechatronics from the Johannes Kepler University, Linz, Austria, in 2001, and the Dr.-Ing. degree in control engineering from Saarland University, Saarbrücken, Germany, in 2005.

From 2001 to 2007, he was with the Institute of Robotics and Mechatronics, German Aerospace Centre (DLR), Wessling, Germany. From May 2007 to June 2009, he was Project Assistant Professor with the Department of Mechano-Informatics, University of Tokyo, Tokyo, Japan. Since then, he has been a Team Leader of the Helmholtz Young Investigators Group for “Dynamic Control of Legged Humanoid Robots” with DLR. In January 2014, he became the Head of the Department of Analysis and Control of Advanced Robotic Systems, DLR. His research interests include nonlinear robot control, flexible joint robots, impedance control, and humanoid manipulation.



Alin Albu-Schäffer received the graduate degree in electrical engineering from the Technical University of Timișoara, Timișoara, Romania, in 1993, and the Ph.D. degree in automatic control from the Technical University of Munich (TUM), Munich, Germany, in 2002.

Since 2012, he has been the Head the Institute of Robotics and Mechatronics, German Aerospace Centre (DLR), Wessling, Germany, which he joined in 1995 as a Ph.D. candidate. He is also a Professor with TUM, holding the Chair for “Sensorbased Robotic Systems and Intelligent Assistance Systems” with the Computer Science Department. His research interests include robot design, modeling, and control, flexible joint and variable compliance robots for manipulation and locomotion, physical human–robot interaction, and bioinspired robot design.