Biologically Inspired Deadbeat control for running: from human analysis to humanoid control and back

Johannes Englsberger, Paweł Kozłowski, Christian Ott, Alin Albu-Schäffer

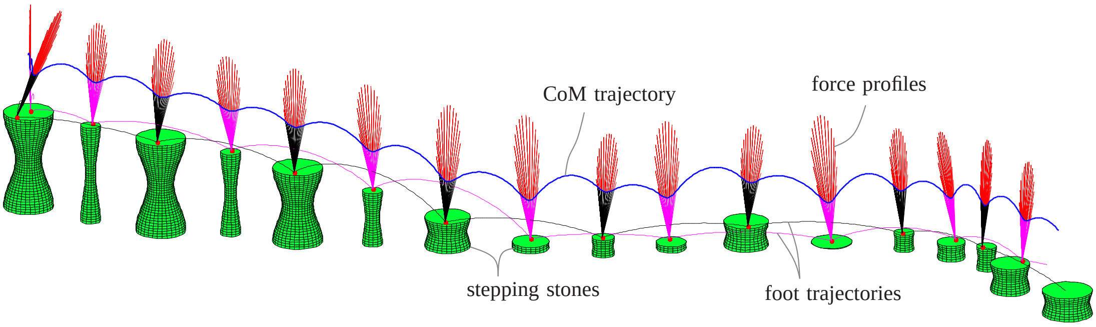

Figure 1: Bipedal point-mass model running on 3D stepping stones based on Biologically Inspired Deadbeat (BID) control.

***Abstract*—This paper works towards 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 was introduced in [1] and enhanced in [2] to not only allow 3D running on flat ground but also on 3D stepping stones. Further contributions of [2] include explicit foot step targeting during running, leg cross-over avoidance and the embedding of BID control into a QP-based whole-body controller. The controller is based on the encoding of leg forces and CoM trajectories during stance as polynomial splines, allowing for intuitive and primarily analytical controller design. It allows a real-time implementation, is highly robust against perturbations and enables versatile running patterns. This paper combines and complements the methods derived in [1] and [2]. It expatiates on the analytical foot-step targeting method introduced in [2], 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 (GRF) to data from human running experiments.**

***Index Terms*—Human running, bipedal robots, running control, biologically inspired, deadbeat, 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 to wheel-based locomotion. Overcoming a set of stepping stones, as shown in Fig. 1, is one possible example where 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 et al. [19] present a SLIP-based two-level controller for running simulations of the ATRIAS robot. Carver et al. [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 et al. [21] introduce bio-inspired swing leg control for running on ground with unexpected height disturbances. Wu et al. [22] present a deadbeat controller for the 3D SLIP model that can cope with unknown ground height variations of up to 30% of the leg length. It is based on multi-dimensional look-up 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

The authors are with Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Wessling, Germany. E-mail: johannes.englsberger@dlr.de

SLIP model and achieve robust running. Recently, Wensing and Orin [24], [25] computed periodic trajectories of the 3D-SLIP offline and applied a linearized control law to stabilize the virtual SLIP model around the periodic solutions. 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 look-up tables and the linearized SLIP feed-back controller is only capable of limited acceleration rates. Also recently, Park et al. [26] presented quadrupedal galloping with the MIT Cheetah 2 based on impulse control. They used 3-rd 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 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 cross-over (especially when running in sharp turns). In [2], we extended the original method to achieve precise foot placement and running on 3D stepping stones (see Fig. 1). The precise foot placement now enables explicit leg cross-over avoidance. Another contribution was the embedding of our running controller into a 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 basic tool for future movement science studies.

The 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 facilitates 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, XI and XII compare the BID control outputs to human experiments, discuss the proposed controller's assets and limitations and conclude the paper.

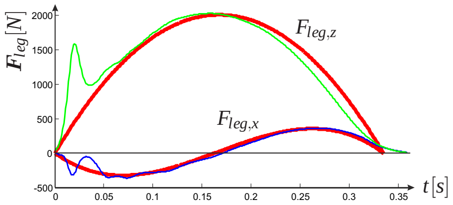

Figure 2: Comparison of experimentally measured human leg forces (blue/green) and polynomial approximations (red).

II. Human running experiments as motivation

The main idea in this paper is to *design desired CoM trajectories* that produce *approximately natural GRF profiles* while fulfilling several *boundary conditions*. It is well known that some physical template models, such as the SLIP, generate ground reaction forces (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 ($\boldsymbol{F}_{leg}$, equivalent to GRF). Figure 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 polynomials [1]. The total force $\boldsymbol{F}_{CoM}$ acting on the CoM can be computed from the leg force $\boldsymbol{F}_{leg}$ and the gravitational force $\boldsymbol{F}_g$ via

$$\boldsymbol{F}_{CoM} = \boldsymbol{F}_{leg} + \boldsymbol{F}_g = \boldsymbol{F}_{leg} + m\,\boldsymbol{g} \quad . \tag{1}$$

Here, m is the robot's total mass and $\boldsymbol{g} = [0\; 0\; -g]^T$ denotes the gravitational acceleration vector. The constant offset between $\boldsymbol{F}_{CoM}$ and $\boldsymbol{F}_{leg}$ in (1) and Newton's 2nd law (CoM acceleration $\ddot{\boldsymbol{x}} = \frac{\boldsymbol{F}_{CoM}}{m}$) motivate us to use - during stance - a 4th order polynomial to encode the vertical CoM position z and 5th order polynomials to encode the horizontal CoM positions x and y, as this correlates to 2nd and 3rd order polynomials for the CoM accelerations $\ddot{x}$, $\ddot{y}$, $\ddot{z}$ and thus leg forces. This polynomial encoding can be written as:

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

Here, $\boldsymbol{t}_{\sigma}^T(t)$, $\boldsymbol{t}_{\dot{\sigma}}^T(t)$ and $\boldsymbol{t}_{\ddot{\sigma}}^T(t)$ denote the time-mapping row vectors that - for a given time t - map the polynomial parameter vectors $\boldsymbol{p}_{\sigma}$ 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 BID 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 heights $\Delta z_{apex,des}$ and $\Delta z_{TD,des}$ 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 touch-down (TD) is supposed to happen. $z_{floor,i}$ denotes the height level of the i-th step. Another design parameter, used in this work, 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 $z_{TD}=z_{floor,i}+\Delta z_{TD}$ 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. CoM 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 $\boldsymbol{x}(t)=[x(t), y(t), z(t)]^T$ and velocity $\dot{\boldsymbol{x}}(t)=[\dot{x}(t), \dot{y}(t), \dot{z}(t)]^T$ can be computed as

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

$$\dot{\boldsymbol{x}}(t)=\dot{\boldsymbol{x}}_0+\boldsymbol{g}\, t \; , \tag{4}$$

where $\boldsymbol{x}_0$ and $\dot{\boldsymbol{x}}_0$ are the initial CoM position and velocity. 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 $\dot{z}_{apex}=0$. 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

$$\Delta t_{apex}=\frac{\dot{z}}{g} \quad . \tag{5}$$

If Δt_{apex} is negative (true for $\dot{z}<0$), then the CoM is already on the descending path of the ballistic flight curve and thus the time of apex is in the past. In the same way, we find the remaining time until touch-down (TD) as

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

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

$$\begin{bmatrix} \boldsymbol{x}_{TD} \\ \dot{\boldsymbol{x}}_{TD} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}+\Delta t_{TD}\,\dot{\boldsymbol{x}}+\frac{\Delta t_{TD}^2}{2}\,\boldsymbol{g} \\ \dot{\boldsymbol{x}}+\Delta t_{TD}\,\boldsymbol{g} \end{bmatrix} \quad . \tag{7}$$

In this work, the relative touch-down height is computed as

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

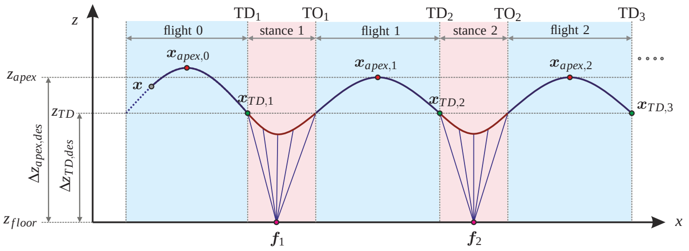

Figure 3: Preview of upcoming flight and stance phases (planar sketch) - used for design of boundary conditions. For readability, a constant floor height z_{floor} is shown here.

That way, nominally the desired relative touch-down height $\Delta z_{TD,des}$ (one of our *design variables*) is achieved, while for challenging initial conditions or perturbations a minimum height difference between apex and touch-down $\Delta_{apex,TD,min}$ 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 4th order polynomial, i.e. it has 5 polynomial parameters. These can be derived using 5 boundary conditions. Fig. 3 graphically displays the used preview of upcoming flight and stance sequences and the corresponding boundary conditions. In this work, - for each previewed contact phase - we make use of four linear vertical boundary conditions that can be combined as

$$\underbrace{\begin{bmatrix} z_{TD,i} \\ \dot{z}_{TD,i} \\ -g \\ -g \end{bmatrix}}_{\boldsymbol{b}_{z,i}} = \underbrace{\begin{bmatrix} \boldsymbol{t}_z^T(0) \\ \boldsymbol{t}_{\dot{z}}^T(0) \\ \boldsymbol{t}_{\ddot{z}}^T(0) \\ \boldsymbol{t}_{\ddot{z}}^T(T_{s,i}) \end{bmatrix}}_{\boldsymbol{B}_{z,i}} \boldsymbol{p}_{z,i} \quad . \tag{9}$$

Here, i denotes the index of the considered step and $\boldsymbol{b}_{z,i}$, $\boldsymbol{B}_{z,i}$ and $\boldsymbol{p}_{z,i}$ denote the corresponding boundary condition vector, boundary condition mapping matrix and vertical polynomial parameter vector, respectively. The first two elements in $\boldsymbol{b}_{z,i}$ imply that CoM position and velocity at the beginning of stance equal the CoM touch-down 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 $\boldsymbol{B}_{z,i}\,\boldsymbol{p}_{z,i}=\boldsymbol{b}_{z,i}$ is

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

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

$$\boldsymbol{r}_{z,i}=\begin{bmatrix} -\boldsymbol{B}_{z,i,square}^{-1}\,\boldsymbol{B}_{z,i,final} \\ 1 \end{bmatrix} \quad , \tag{11}$$

where $\boldsymbol{B}_{z,i,final}$ is the last column in $\boldsymbol{B}_{z,i}$, while $\boldsymbol{B}_{z,i,square}$ 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

$z_{apex,i}$ of the CoM during the i-th upcoming flight phase (see Fig. 3). The vertical CoM state of the i-th take-off (at end of i-th stance time $T_{s,i}$) can be computed via (2) as

$$z_{TO,i} = \boldsymbol{t}_z^T(T_{s,i})\ \boldsymbol{p}_{z,i} \tag{12}$$

$$\dot{z}_{TO,i} = \boldsymbol{t}_{\dot{z}}^T(T_{s,i})\ \boldsymbol{p}_{z,i} \quad . \tag{13}$$

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

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

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

$$z_{apex,i,des} = z_{floor,i+1} + \Delta z_{apex,des} \ . \tag{15}$$

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

$$\begin{aligned} 0 \ &= \ \frac{t_{\dot{z}}^T \boldsymbol{r}_{z,i}}{2g}\ \tilde{p}_{z,i}^2 + (t_z^T \boldsymbol{r}_{z,i} + \frac{t_{\dot{z}}^T\, p_{z,i,0}\ \ t_{\dot{z}}^T \boldsymbol{r}_{z,i}}{g})\ \tilde{p}_{z,i} + \\ &+ \ \frac{(t_{\dot{z}}^T\, p_{z,i,0})^2}{2g} - z_{apex,i,des} \end{aligned} \tag{16}$$

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

$$\begin{aligned} \tilde{p}_{z,i} \ &= \ \frac{2\, \dot{z}_{TD,i} - gT_{s,i} - \Gamma}{4\, T_{s,i}^3} \ , \\ \Gamma \ &= \ \sqrt{g(gT_{s,i}^2 - 4\, \dot{z}_{TD,i}\ T_{s,i} + 8\ (z_{apex,i,des} - z_{TD,i}))} \end{aligned} \tag{17}$$

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

B. Horizontal planning and boundary conditions

In this work, 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:

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

Here, $\boldsymbol{b}_{\chi,i}$, $\boldsymbol{B}_{\chi,i}$ and $\boldsymbol{p}_{\chi,i}$ denote the horizontal boundary condition vector, boundary condition mapping matrix and polynomial parameter vector, respectively. As in Sec. V-A, the first two elements of $\boldsymbol{b}_{\chi,i}$ imply that the initial CoM state is equal to the CoM touch-down 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 intermediate control target - specifies the horizontal CoM touch-down position $\chi_{TD,i+1,des}$ of the upcoming step. Since - in case of no perturbations - the horizontal velocity during flight is constant, we can propagate the take-off state to each upcoming touch-down position via

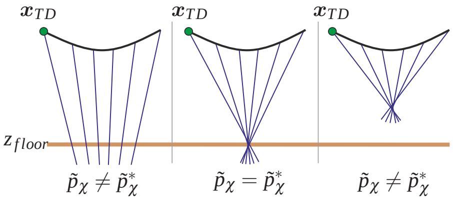
$\boldsymbol{x}_{TD}$
$\boldsymbol{x}_{TD}$
$\boldsymbol{x}_{TD}$
z_{floor}
$\tilde{p}_\chi \neq \tilde{p}_\chi^*$
$\tilde{p}_\chi = \tilde{p}_\chi^*$
$\tilde{p}_\chi \neq \tilde{p}_\chi^*$

Figure 4: Effect of $\tilde{p}_\chi$ on force ray focusing (lines of action).

$$\chi_{TD,i+1,des} = \chi_{TO,i} + T_{f,i}\, \dot{\chi}_{TO,i} = (\boldsymbol{t}_{\chi,i}^T(T_{s,i}) + T_{f,i}\, \boldsymbol{t}_{\dot{\chi}}^T(T_{s,i}))\, \boldsymbol{p}_{\chi,i} \quad , \tag{19}$$

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

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

The nullspace base vector $\boldsymbol{r}_{\chi,i}$ 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 point $\boldsymbol{x}_{int,i} = [x_{int,i}, y_{int,i}, z_{floor,i}]$ of the leg force with the floor and minimize the integral of the mean square deviation from its mean value $\overline{x}_{int,i}$. For a given time in the i-th stance $t_s \in [0, T_{s,i}]$, it's horizontal components are

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

Here, $f_{leg,\chi,i}(t_s)$ and $f_{leg,z,i}(t_s)$ are the horizontal and vertical components of the leg force $\boldsymbol{F}_{leg,i}$ and $z(t_s)$ is the height of the CoM. The horizontal components of the mean intersection point $\overline{\boldsymbol{x}}_{int,i} = [\overline{x}_{int,i}, \overline{y}_{int,i}, z_{floor,i}]$ can be computed via

$$\overline{\chi}_{int,i} = \frac{1}{T_{s,i}} \int_{t_s=0}^{T_{s,i}} \chi_{int,i}(t_s)\, dt_s = \underbrace{\frac{1}{T_{s,i}} \int_{t_s=0}^{T_{s,i}} \boldsymbol{d}_{\chi,i}^T(t_s)\, dt_s}_{\boldsymbol{e}_{\chi,i}^T}\ \boldsymbol{p}_{\chi,i} \ . \tag{22}$$

Here, $\boldsymbol{e}_{\chi,i}^T$ is a constant row vector. The deviation of the i-th time-varying intersection point from its mean value is

$$\Delta\chi_{int,i}(t_s) = \chi_{int,i}(t_s) - \overline{\chi}_{int,i} = \underbrace{(\boldsymbol{d}_{\chi,i}^T(t_s) - \boldsymbol{e}_{\chi,i}^T)}_{\boldsymbol{k}_{\chi,i}^T(t_s)}\ \boldsymbol{p}_{\chi,i} \quad . \tag{23}$$

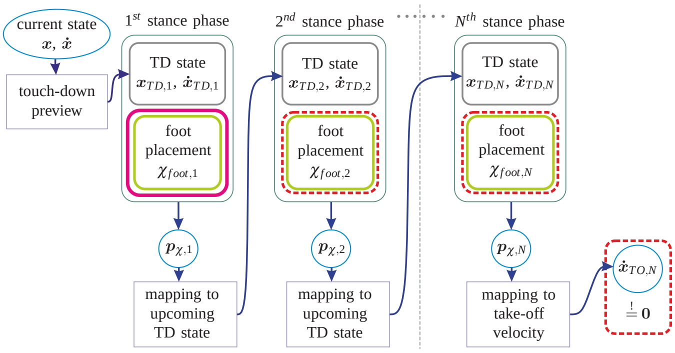

Figure 5: CoM touch-down adjustment for foot targeting.

The square of the deviation at a given time t_s is

$$\Delta\chi^2_{int,i}(t_s) = \boldsymbol{p}^T_{\chi,i}\,\underbrace{\boldsymbol{k}_{\chi,i}(t_s)\,\boldsymbol{k}^T_{\chi,i}(t_s)}_{\boldsymbol{L}_{\chi,i}(t_s)}\,\boldsymbol{p}_{\chi,i}\ . \qquad (24)$$

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

$$\chi_{int,i,ms} = \boldsymbol{p}^T_{\chi,i}\,\underbrace{\frac{1}{T_{s,i}}\int_{t_s=0}^{T_{s,i}}\boldsymbol{L}_{\chi,i}(t_s)\,dt_s}_{\boldsymbol{M}_{\chi,i}}\,\boldsymbol{p}_{\chi,i} = \boldsymbol{r}^T_{\chi,i}\boldsymbol{M}_{\chi,i}\,\boldsymbol{r}_{\chi,i}\,\tilde{p}^2_{\chi,i} + \\ +2\,\boldsymbol{r}^T_{\chi,i}\boldsymbol{M}_{\chi,i}\,\boldsymbol{p}_{\chi,i,0}\,\tilde{p}_{\chi,i} + \boldsymbol{p}^T_{\chi,i,0}\,\boldsymbol{M}_{\chi,i}\,\boldsymbol{p}_{\chi,i,0}\ . \qquad (25)$$

Due to the nonlinearity of (21), solving for $\boldsymbol{M}_{\chi,i}$ analytically is computationally expensive. Instead, $\boldsymbol{d}^T_{\chi,i}$, $\boldsymbol{e}^T_{\chi,i}$, $\boldsymbol{k}^T_{\chi,i}$, $\boldsymbol{L}_{\chi,i}$ and $\boldsymbol{M}_{\chi,i}$ are approximated numerically by evaluating the above equations for n_{approx} time samples equally spread along the stance period. That way, the integrals turn into sums, which highly facilitates computation. We found that $n_{approx} = 10$ yields sufficient accuracy. Now, with the approximate matrix $\boldsymbol{M}_{\chi,i,approx}$ and differentiating (25) with respect to $\tilde{p}_{\chi,i}$, we find the optimal parameter

$$\tilde{p}^*_{\chi,i} = -\frac{\boldsymbol{r}^T_{\chi,i}\ \boldsymbol{M}_{\chi,i,approx}\ \boldsymbol{p}_{\chi,i,0}}{\boldsymbol{r}^T_{\chi,i}\ \boldsymbol{M}_{\chi,i,approx}\ \boldsymbol{r}_{\chi,i}}\ , \qquad (26)$$

which minimizes the mean square deviation as defined above. With (26), (20) turns into

$$\boldsymbol{p}_{\chi,i} = \underbrace{(\boldsymbol{I} - \frac{\boldsymbol{r}_{\chi,i}\ \boldsymbol{r}^T_{\chi,i}\ \boldsymbol{M}_{\chi,i,approx}}{\boldsymbol{r}^T_{\chi,i}\ \boldsymbol{M}_{\chi,i,approx}\ \boldsymbol{r}_{\chi,i}})}_{\boldsymbol{\Omega}_{\chi,i}}\ \underbrace{\boldsymbol{B}^T_{\chi,i}(\boldsymbol{B}_{\chi,i}\,\boldsymbol{B}^T_{\chi,i})^{-1}}_{\boldsymbol{B}^+_{\chi,i}}\ \boldsymbol{b}_{\chi,i}\ , \qquad (27)$$

which directly maps the horizontal boundary conditions $\boldsymbol{b}_{\chi,i}$ to appropriate polynomial parameter vectors $\boldsymbol{p}_{\chi,i}$ (including best force focus). If - as in [1] - horizontal CoM touch-down target positions (or similarly: take-off velocities) are used as boundary conditions, (27) provides the solution to the problem.

C. Foot step targeting and leg cross-over avoidance

In [1], the resulting foot positions could not be controlled directly, which caused problems with leg cross-over (see Fig. 7). Also when precise foot placement is required - for example 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 $\overline{\chi}_{int,i} = \chi_{foot,i}$ and $\boldsymbol{e}^T_{\chi,i} = \boldsymbol{e}^T_{\chi,i,approx}$ in (22), and inserting (27), we can solve for the desired upcoming CoM touch-down position[1] $\chi_{TD,i+1,des}$, which corresponds to the desired foot location $\chi_{foot,i}$. Re-substituting this particular $\chi_{TD,i+1,des}$ in (27) finally yields

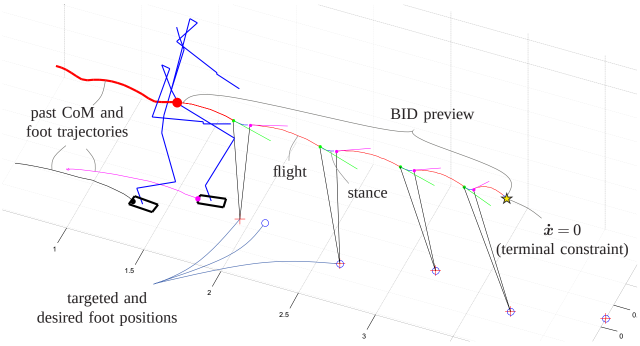

Figure 6: BID preview of Toro (displayed as stickman) running in OpenHRP. All desired foot positions (except the first one) are previewed to be perfectly tracked.

$$\boldsymbol{p}_{\chi,i} = [\underbrace{(\boldsymbol{I}\ -\ \boldsymbol{e}^{\oplus}_{\chi,i}\,\boldsymbol{e}^T_{\chi,i,approx})\ \ \boldsymbol{\Omega}_{\chi,i}\,\boldsymbol{\Pi}_{\chi,i},}_{\boldsymbol{A}_{TD,\chi,i}}\quad \boldsymbol{e}^{\oplus}_{\chi,i}]\begin{bmatrix}\chi_{TD,i}\\ \dot{\chi}_{TD,i}\\ \chi_{foot,i}\end{bmatrix}\ . \qquad (28)$$

Here, $\boldsymbol{A}_{TD,\chi,i}$ maps the i-th touch-down state to $\boldsymbol{p}_{\chi,i}$ and the specific pseudo-inverse $\boldsymbol{e}^{\oplus}_{\chi,i} = \frac{\boldsymbol{\Omega}_{\chi,i}\ \boldsymbol{\pi}_{\chi,i}}{\boldsymbol{e}^T_{\chi,i,approx}\ \boldsymbol{\Omega}_{\chi,i}\ \boldsymbol{\pi}_{\chi,i}}$ of $\boldsymbol{e}_{\chi,i,approx}$ maps the i-th foot position. The matrix $\boldsymbol{\Pi}_{\chi,i}$ combines the first two column vectors of $\boldsymbol{B}^+_{\chi,i}$, while $\boldsymbol{\pi}_{\chi,i}$ is its final column. Note: the third and fourth boundary conditions in (18) are implicitly accounted for. We will now use all previewed desired footholds $\chi_{foot,i} = \chi_{foot,des,i}$ (except the first one) and the final take-off velocity $\dot{\chi}_{TO,N} = \boldsymbol{0}$ as constraints and solve for the first foothold $\chi_{foot,1}$ (*control variable*) and all future horizontal polynomial parameter vectors $\boldsymbol{p}_{\chi,i}$, which yield perfect tracking of the future desired footholds. By combining the touch-down state into $\boldsymbol{\tau}_{\chi,i} = [\chi_{TD,i}, \dot{\chi}_{TD,i}]^T$, (28) becomes

$$\boldsymbol{p}_{\chi,i} = \boldsymbol{A}_{TD,\chi,i}\ \ \boldsymbol{\tau}_{\chi,i} +\ \boldsymbol{e}^{\oplus}_i\ \ \chi_{foot,i}\ . \qquad (29)$$

Each upcoming touch-down state can be previewed as

$$\boldsymbol{\tau}_{\chi,i+1} = \begin{bmatrix}\chi_{TD,i+1}\\ \dot{\chi}_{TD,i+1}\end{bmatrix} = \underbrace{\begin{bmatrix}\boldsymbol{t}^T_{\chi,i}(T_{s,i}) + T_{f,i}\ \dot{\boldsymbol{t}}^T_{\chi}(T_{s,i})\\ \dot{\boldsymbol{t}}^T_{\chi}(T_{s,i})\end{bmatrix}}_{\boldsymbol{S}_i}\ \boldsymbol{p}_{\chi,i}\ . \qquad (30)$$

Starting from $i = 1$ and propagating (29) and (30) forward, we find the following expression for the polynomial parameter vector of the N-th (i.e. final) previewed stance phase:

$$\boldsymbol{p}_{\chi,N} = \boldsymbol{G}_{\chi,1}\ \ \boldsymbol{A}_{TD,\chi,1}\ \ \boldsymbol{\tau}_{\chi,1}\ +\sum_{i=1..N}(\boldsymbol{G}_{\chi,i}\ \ \boldsymbol{e}^{\oplus}_{\chi,i}\ \ \chi_{foot,i})\ . \qquad (31)$$

[1] This is why $\chi_{TD,i+1,des}$ was called an "intermediate control target" earlier.

The matrices $\boldsymbol{G}_{\chi,i} = \boldsymbol{G}_{\chi,i+1}\,\boldsymbol{A}_{TD,\chi,i+1}\,\boldsymbol{S}_i$ are evaluated by starting with $\boldsymbol{G}_{\chi,N} = \boldsymbol{I}_{6x6}$ and iterating backwards until $i = 1$. Now, with (2) and (31), we find the horizontal take-off velocity after the final previewed stance phase as

$$\dot{\chi}_{TO,N} = \boldsymbol{t}_{\dot{\chi}}^T(T_{s,N})\ \boldsymbol{p}_{\chi,N}\ . \qquad (32)$$

To guarantee stability, we choose $\dot{\chi}_{TO,N} \overset{!}{=} 0$ 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 $\dot{z}_{TO,N} \overset{!}{=} 0$ 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 (31)-(32) for $\chi_{foot,1}$ which finally yields the sought-after first foot placement

$$\chi_{foot,1} = \frac{-\boldsymbol{t}_{\dot{\chi}}^T(T_{s,N})\,(\boldsymbol{G}_{\chi,1}\,\boldsymbol{A}_{TD,\chi,1}\,\boldsymbol{\tau}_{\chi,1}+\boldsymbol{\eta})}{\boldsymbol{t}_{\dot{\chi}}^T(T_{s,N})\,\boldsymbol{G}_{\chi,1}\,\boldsymbol{e}_{\chi,1}^{\oplus}}\ , \qquad (33)$$

$$\boldsymbol{\eta} = \sum_{i=2..N} (\boldsymbol{G}_{\chi,i}\ \boldsymbol{e}^{\oplus}_{\chi,i}\ \chi_{foot,des,i})\ .$$

Now we solve for all horizontal polynomial parameter vectors $\boldsymbol{p}_{\chi,i}$ by alternately evaluating (29) and (30). As foot positions in (29), we use $\chi_{foot,i} \in \{\chi_{foot,1}, \chi_{foot,des,2}, \cdots \chi_{foot,des,N}\}$. During stance we freeze the first foot position $\chi_{foot,1}$ and polynomial parameter vector $\boldsymbol{p}_{\chi,1}$ (feed-forward) and use the second foothold $\chi_{foot,2}$ as control variable instead. That way - even in face of unknown perturbations - the foot targets are continuously adjusted. Equation (33) is adopted accordingly.

One feature of our framework is that due to the multi-step 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 work, we implemented the foot trajectories as polynomials. The achieved precise foot targeting is particularly interesting for running over 3D stepping stones or other restricted surfaces.

An additional feature of precise foothold targeting is that leg cross-over 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, i.e. two for each foot. The method adjusts the second/third previewed 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.

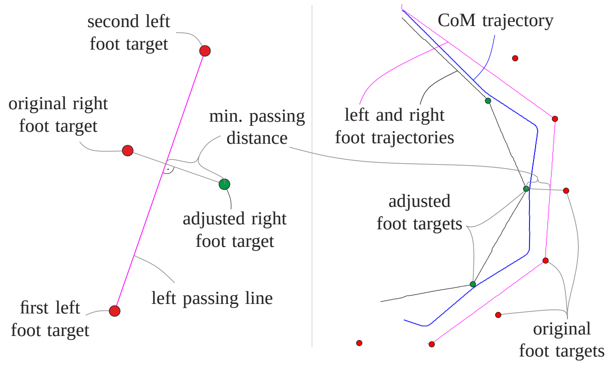

Figure 7: Leg cross-over avoidance,
left: scheme (depicted for left pass), right: simulation output

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 pre-planning a whole sequence of upcoming steps would be sufficient. Yet, to cope with perturbations, we propose a state feed-back control method, which is based on continuous re-planning of the desired contact forces throughout both flight and stance phases. To this end, during flight the previewed CoM touch-down state is updated (see Fig. 8) by inserting (6) in (3) and (4). In contrast to [1] (no feedback during stance), during stance, the first take-off state is predicted via

$$\begin{bmatrix}\sigma_{TO,1}\\ \dot{\sigma}_{TO,1}\end{bmatrix} = \underbrace{\begin{bmatrix}\sigma\\ \dot{\sigma}\end{bmatrix}}_{feedback} + \underbrace{\begin{bmatrix}\boldsymbol{t}_{\sigma}^T(T_{s,1}) - \boldsymbol{t}_{\sigma}^T(t_s)\\ \boldsymbol{t}_{\dot{\sigma}}^T(T_{s,1}) - \boldsymbol{t}_{\dot{\sigma}}^T(t_s)\end{bmatrix}\boldsymbol{p}_{\sigma,1}}_{preview}\ , \quad \sigma \in \{x,y,z\} \qquad (34)$$

Here, $\boldsymbol{t}_{\sigma}^T(t)$ and $\boldsymbol{t}_{\dot{\sigma}}^T(t)$ 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 $\boldsymbol{p}_{\sigma,1}$) is applied. This offset is added to the current measured state to predict the take-off state, which in turn is used to compute the upcoming CoM touch-down state. Note: after touch-down, the force profile of the current stance phase is frozen and commanded to the robot as feed-forward. 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: during flight, the first upcoming foot position is one of the main *control inputs* (see (33)). Whilst 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.

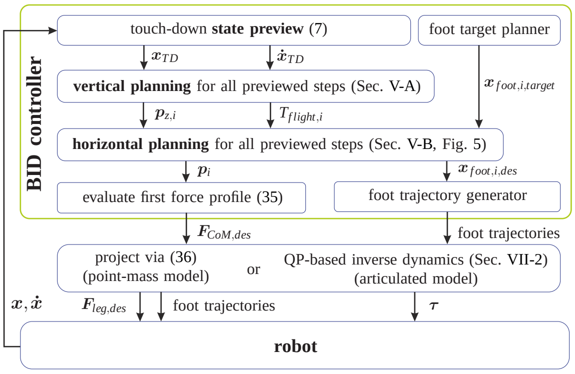
BID controller
touch-down **state preview** (7)
foot target planner
$\boldsymbol{x}_{TD}$
$\dot{\boldsymbol{x}}_{TD}$
vertical planning for all previewed steps (Sec. V-A)
$\boldsymbol{x}_{foot,i,target}$
$\boldsymbol{p}_{z,i}$
$T_{flight,i}$
horizontal planning for all previewed steps (Sec. V-B, Fig. 5)
$\boldsymbol{p}_i$
$\boldsymbol{x}_{foot,i,des}$
evaluate first force profile (35)
foot trajectory generator
$\boldsymbol{F}_{CoM,des}$
foot trajectories
project via (36) (point-mass model)
or
QP-based inverse dynamics (Sec. VII-2) (articulated model)
$\boldsymbol{x}, \dot{\boldsymbol{x}}$
$\boldsymbol{F}_{leg,des}$
foot trajectories
$\boldsymbol{\tau}$
robot

Figure 8: Computation flow of BID controller (outline).

VII. Guaranteeing feasibility

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

$$\boldsymbol{F}_{CoM,des}(t_s) = m \begin{bmatrix} \boldsymbol{t}_{\ddot{x}}^T(t_s)\,\boldsymbol{p}_{x,1} \\ \boldsymbol{t}_{\ddot{y}}^T(t_s)\,\boldsymbol{p}_{y,1} \\ \boldsymbol{t}_{\ddot{z}}^T(t_s)\,\boldsymbol{p}_{z,1} \end{bmatrix} \quad , \tag{35}$$

i.e. the polynomial of the first force profile is evaluated. The corresponding desired leg force $\boldsymbol{F}_{leg,des}$ 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.

1) 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 constraint to point along the unit vector $\boldsymbol{u}_{x,f}$ pointing from CoM to point foot. As the other two spatial directions are unactuated, the desired leg force $\boldsymbol{F}_{leg,des}$ has to be projected to the feasible direction[2]:

$$\boldsymbol{F}_{leg,f} = \boldsymbol{u}_{x,f}\,\boldsymbol{u}_{x,f}^T\,\boldsymbol{F}_{leg,des} \quad . \tag{36}$$

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

2) Articulated multi-body 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 multi-body models such as simulated or real humanoid robots. To this end, we use an inverse dynamics based whole-body control framework similar to [24], [27]. It solves a single quadratic program (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 (35) 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 only relates to ground reaction wrenches and joint torques, whilst stability or balance (depending on the physical limitations of the robot at hand) is not guaranteed.

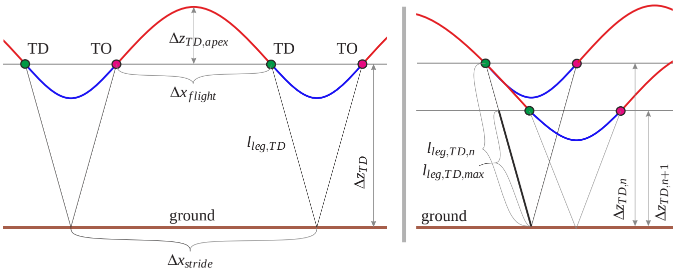

Figure 9: Correlations for stationary running.

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 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: 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 corresponding leg length in a multi-body model. This is an approximation of course.

A. Nominally feasible gait design

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

$$T_f = \sqrt{\frac{8\,\Delta z_{TD,apex}}{g}} \quad . \tag{37}$$

Here, $\Delta z_{TD,apex}$ denotes the height difference between apex and touch-down. 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

$$T_f = f_{flight}\,\underbrace{(T_s+T_f)}_{\frac{\Delta x_{stride}}{v_{mean}}} \;\leq\; f_{flight}\,\frac{\Delta x_{stride,max}}{v_{mean}} \quad . \tag{38}$$

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 (37) and (38), we find a condition for the maximum allowable height difference

[2] Note: for more complex robots this projection may not be necessary.

between apex and touch-down

$$\Delta z_{TD,apex} \leq \underbrace{\frac{g}{8}\,(f_{flight}\,\frac{\Delta x_{stride,max}}{v_{mean}})^2}_{\Delta z_{TD,apex,max}} \quad . \tag{39}$$

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

$$\Delta z_{TD} \leq \underbrace{\sqrt{l^2_{leg,TD,max} - (\frac{\Delta\hat{x}_{stride} - \Delta\hat{x}_{flight}}{2})^2}}_{\Delta z_{TD,max}} \quad . \tag{40}$$

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 (39) and touch-down height difference (40), the nominal desired touch-down and apex height difference (above each upcoming floor height) become

$$\Delta z_{TD,des} = min(\Delta z_{TD,nominal},\ \Delta z_{TD,max}) \qquad and \tag{41}$$

$$\Delta z_{apex,des} = \Delta z_{TD,des} + min(\Delta z_{TD,apex,nominal},\ \Delta z_{TD,apex,max}) \quad ,$$

where $\Delta z_{TD,nominal}$ and $\Delta z_{TD,apex,nominal}$ act as upper limits. Also, the nominal stance time can be computed with (37) as

$$T_s = \frac{1 - f_{flight}}{f_{flight}} \sqrt{\frac{8\,\Delta z_{TD,apex}}{g}} \quad . \tag{42}$$

That way, the design parameters introduced in Sec. 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 touch-down leg length $l_{leg,TD,max}$ and mean horizontal speed v_{mean} to maximize kinematic feasibility.

B. Active leg length control

In 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 multi-body robots. To ease this problem, we adjust the original BID plan via the following iteration scheme (see Fig. 9 (right))

$$\Delta z_{TD,i,n+1} = min(\Delta z_{TD,desired}\ ,\ \frac{l_{leg,TD,max}}{l_{leg,TD,i,n}}\,\Delta z_{TD,i,n})) \quad . \tag{43}$$

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

IX. Simulations and evaluation of BID control

A. BID-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. Figure 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 setup in the following way: no perturbation during the first 4 seconds, then 4 seconds of constant force acting (magnitude: -50N (corresponding to $\approx$ 10% of the robot's mass (here 50kg)), force direction: purely x, y and z, respectively), followed by 4 seconds 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 $\boldsymbol{x}_{joystick} = [x_{joystick}, y_{joystick}]^T$, 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: the perturbation forces in the shown 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. It has to be mentioned, that for extreme perturbations, the leg length could grow to unrealistic levels (due to the constant touch-down height). To assure leg length feasibility, the method from Sec. VIII-B can be applied. The controller is most sensitive against strong unknown perturbations that point towards the ground. Here, the maximum permanent force the controller could withstand in simulation was $-750N$, i.e. 1.5 times the robots weight. For higher forces, the robot's CoM would hit the ground.

Figure 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, whereas in 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.

Figure 12 shows how far the force intersection point $\chi_{int}(t_s)$ deviates from the mean intersection point $\overline{\chi}_{int}$ (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 $2\frac{m}{s}$. The stance time is set to $150ms$. The initial range of deviation is about $22mm$, while for stationary running it is about $6mm$. This shows that the original (non-projecting) method is well applicable for small-footed robots and that (36) typically has minor influence.

B. QP-based multi-body simulations

To proof the applicability of the biologically inspired deadbeat (BID) control framework, we embedded it into the QP-based whole-body controller from Sec. VII-2 and performed full-body simulations of the humanoid robot Toro [29] in

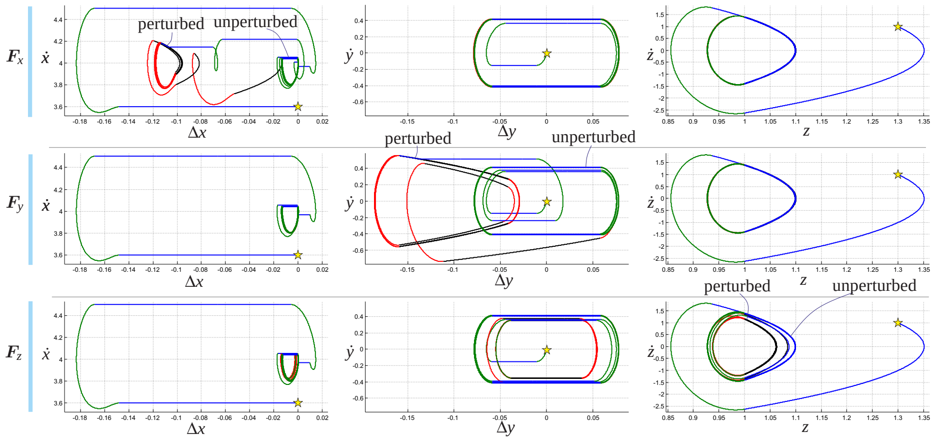

Figure 10: Robustness examination (point-mass under BID control) for different constant external forces. Perturbation inactive: stance green, flight blue. Perturbation active: stance red, flight blue.

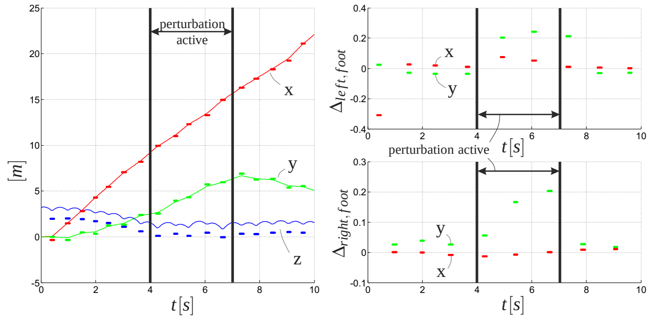

Figure 11: CoM trajectories and foot target tracking performance in nominal and perturbed case (point mass simulation over stepping stones). Perturbing force: $30N$ in y-direction.

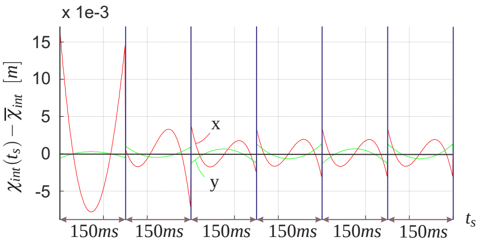

Figure 12: Deviation of force intersection point $\chi_{int}(t_s)$ from mean intersection point $\overline{\chi}_{int}$ in case that (in contrast to other point-mass simulations) the *forces are not projected*. Along the time-axis, the stance phases are pieced together.

OpenHRP [30]. It has to be noted, that Toro's joint torque and velocity limits were omitted in the simulations. Figure 13 shows Toro running at $5m/s$. The gait parameters $\Delta z_{TD,des}$, $\Delta z_{apex,des}$ and T_s were computed via the method from Sec. VIII-A to make such high running speed kinematically feasible for Toro. Following intuitive design parameters (as described in Sec. VIII-A) were chosen: desired flight percentage $f_{flight} = 0.7$, maximum desired stride length $\Delta x_{stride,max} = 1.4\,m$, maximum desired touch-down leg length $l_{leg,TD,max} = 0.86\,m$, nominal touch-down height $\Delta z_{TD,nominal} = 0.86\,m$ and nominal height difference between touch-down 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 $0\,m/s$ 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. The CoM motion (see Fig. 14, colored) follows the desired joystick reference (black) nicely. The actual reference were corresponding foot targets that were derived from the nominal sway-free joystick input. Knowing the vertical previewed dynamics and thus the times to each upcoming touch-down, the foot targets were placed at lateral offsets from the nominal sway-free and continuous reference. A very important quality of a running controller is its reactiveness 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 control framework, we performed multiple simulations where 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 $3\,\frac{m}{s}$ (after ramping up from $0\,\frac{m}{s}$ until second 3). From second $3.5-4.5$ 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 is well able to compensate for these perturbations and recovers after just a few steps. The steady state error of about $0.1\,m$ in 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 touch-down*, while the continuous joystick reference keeps moving throughout stance. The kinematic feasibility of the

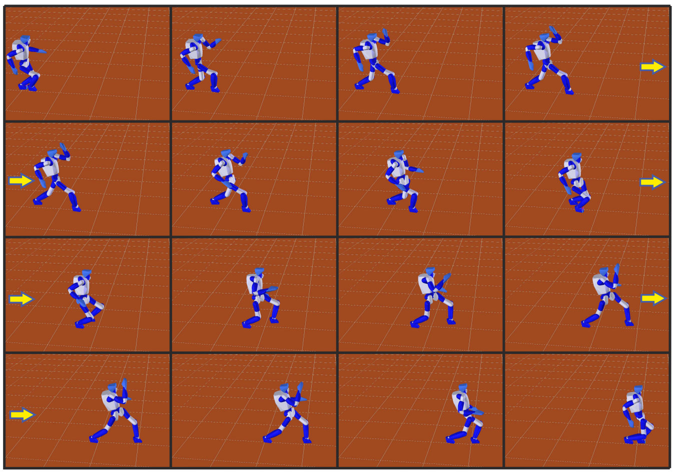

Figure 13: Toro [29] running in OpenHRP [30] at 5 m/s.

running gait under these strong perturbations was facilitated by the methods from Sec. VIII.

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

X. Comparison to 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 arises, how well 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. Figure 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 apparent that the lines of action of the ground reaction forces (GRF) in humans are not as strictly focused as the ones designed in our BID controller (compare to figures 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). Figure 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 to 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.

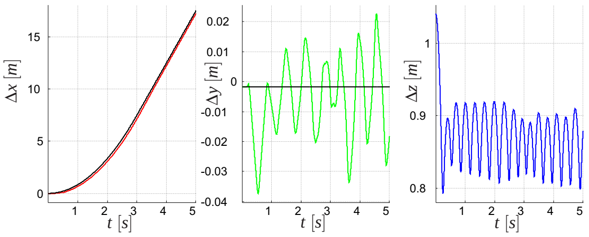

Figure 14: Toro's CoM while running in OpenHRP at $0-5\ \frac{m}{s}$

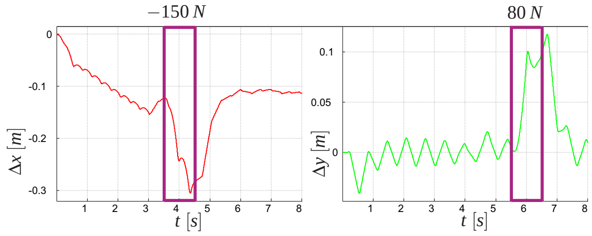

Figure 15: CoM error during push-recovery simulation.

XI. Discussion and outlook

A. Strengths and limitations of current control framework

The proposed control framework can be called a closed-form solution to 3D running. Only the matrices $\boldsymbol{M}_{\chi,i}$ in Sec. V-B have to be approximated numerically but they can then be used for the further analytical derivations such as our analytical foot-step targeting method. The polynomial parameter vectors resulting from the BID controller are analytic and thus very insightful and convenient. The trajectory generation and control method described in this paper yields leg force profiles that are independent of the specific hardware design of a particular robot, i.e. the method is generic. The control framework might be used to identify required actuator characteristics for the design of new robots.

For our simulations, we used a standard PC (3.3 GHz, quad-core, Win7 64bit). 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 (deadbeat control), i.e. the controller is perfectly stable. In case of actuation limits, the control commands may have to be adjusted (e.g. via (36) for point-mass point-feet robots), so 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 will cause perturbations. Anyhow, due to its high robustness in simulations, we expect good performance of the controller.

A drawback of our current control setup when compared to 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 challenging task. Anyway, it has to be tackled in future research to enhance the capabilities of humanoid running and make it more comparable to its natural counterpart.

[3] By "matching" we mean that basic gait parameters such as stance time, apex and touch-down height and also the human subject's and the model's mass have to be aligned. Otherwise a comparison - especially in time domain - would be impossible/useless.

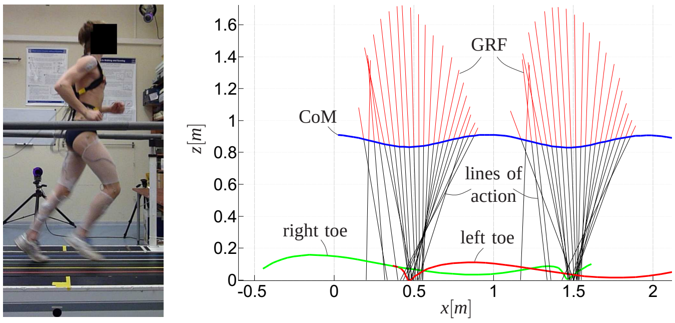

Figure 16: Human running experiment. Left: subject running on force plate treadmill, right: trajectories and GRF.

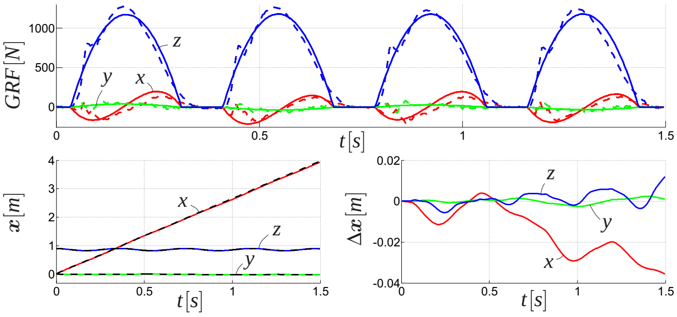

Figure 17: Comparison of human experiment data [31], [32] (dashed) and output of matched BID simulation. top: ground reaction forces (GRF), left: CoM position, right: CoM error.

B. Comparison to other works

When compared to SLIP control, the main feature of our presented BID controller is its analyticity, which allows for explicit solutions, e.g. for three-dimensional 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 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 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 case of limited allowable contact area (see Fig. 18).

C. Potential usage, extensions and future work

One interesting aspect in human running is the center of pressure's (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 desired 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 work, we work with locally flat stepping stones (see figure 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, 3D 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 in detail the Biologically Inspired Deadbeat (BID) controller, a concept for three-dimensional bipedal running. It encodes the leg forces during stance as polynomials. The proposed controller has deadbeat properties, i.e. in the nominal case it reaches the desired boundary conditions after just one stance phase. The controller facilitates agile, precise and versatile running motions and is very robust against external perturbations. It can be utilized to achieve explicit foot targeting and running over three-dimensional stepping stones. Additionally, the paper describes methods for leg cross-over 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 $\frac{m}{s}$ and demonstrated push-recovery. The CoM trajectories and ground reaction forces resulting from BID control were compared to 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.

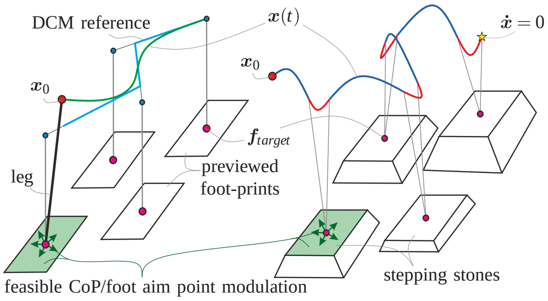

Figure 18: Analogy of DCM (walking) and BID (running); stepping stones are comparable to finite-sized feet.

Acknowledgements

The authors want to cordially thank Christian Rode, Roy Müller and Moritz Maus from the Locomotion Laboratory (Technical University of Darmstadt) and Imperial College London for kindly providing the human running data and the photo used in this work. They also thank Debora Clever from the Optimization in Robotics & Biomechanics group (University of Heidelberg) for providing human experiment data and for the good discussions.

This research is partly supported by the Initiative and Networking Fund of Helmholtz Association through a Helmholtz Young Investigators Group (Grant no. VH-NG-808).

References

[1] J. Englsberger, P. Kozlowski, and C. Ott, "Biologically inspired deadbeat controller for bipedal running in 3d," in *IEEE/RSJ Int. Conf. on Intell. Robots and Systems*, 2015, pp. 989–996.

[2] ——, "Biologically inspired deadbeat control for running on 3d stepping stones," in *IEEE-RAS Int. Conf. on 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," *Mathematical Biosciences*, vol. 15, pp. 1–37, 1972.

[5] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *IEEE Int. Conf. on Robotics and Automation*, 2003, pp. 1620–1626.

[6] P.-B. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2006, pp. 137–142.

[7] S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro, and K. Yokoi, "Biped walking stabilization based on linear inverted pendulum tracking," in *Int. Conf. on Intell. Robots and Systems*, 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 *Int. Conf. on Intell. Robots and Systems*, 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. of Robotics Research*, 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 *IEEE-RAS Int. Conf. on Humanoid Robots*, 2012, pp. 726–733.

[11] J. Englsberger, C. Ott, and A. Albu-Schäffer, "Three-dimensional bipedal walking control based on divergent component of motion," *Robotics, IEEE Transactions on*, vol. 31, no. 2, pp. 355–368, 2015.

[12] "Darpa robotics challenge." [Online]. Available: http://www.theroboticschallenge.org/

[13] M. Raibert, *Legged robots that balance*. The MIT Press,Cambridge, MA, 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 *IEEE Int. Conf. on Robotics and Automation*, 2004, pp. 3189–3194.

[15] T. Takenaka, T. Matsumoto, T. Yoshiike, T. Hasegawa, S. Shirokura, H. Kaneko, and A. Orita, "Real time motion generation and control for biped robot, 4th report: Integrated balance control," in *Int. Conf. on Intell. Robots and Systems*, 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 *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 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 *IEEE-RAS Int. Conf. on Humanoid Robots*, 2014, pp. 930–937.

[18] H. Geyer, A. Seyfarth, and R. Blickhan, "Compliant leg behaviour explains basic dynamics of walking and running," *Proceedings of the Royal Society B: Biological Sciences*, 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 *Int. Conf. on Intell. Robots and Systems*, 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, 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 and Biomimetics*, vol. 8, no. 4, 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," *Robotics, IEEE Transactions on*, vol. 29, no. 5, pp. 1114–1124, Oct 2013.

[23] D. Koepl and J. Hurst, "Impulse control for planar spring-mass running," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 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 *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 5134–5140.

[25] P. Wensing and D. Orin, "3d-slip steering for high-speed humanoid turns," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, 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 *IEEE Int. Conf. on Robotics and Automation*, 2015, pp. 5163–5170.

[27] T. Koolen, J. Smith, G. Thomas, S. Bertrand, J. Carff, N. Mertins, D. Stephen, P. Abeles, J. Englsberger, S. McCrory, J. van Egmond, M. Griffioen, M. Floyd, S. Kobus, N. Manor, S. Alsheikh, D. Duran, L. Bunch, E. Morphis, L. Colasanto, K.-L. H. Hoang, B. Layton, P. Neuhaus, M. Johnson, and J. Pratt, "Summary of team ihmc's virtual robotics challenge entry," in *Int. Conf. on Humanoid Robots*, 2013.

[28] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Auton. Robots*, vol. 35, no. 2-3, pp. 161–176, 2013.

[29] J. Englsberger, A. Werner, C. Ott, B. Henze, M. A. Roa, G. Garofalo, R. Burger, A. Beyer, O. Eiberger, K. Schmid, and A. Albu-Schäffer, "Overview of the torque-controlled humanoid robot toro," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2014, pp. 916–923.

[30] F. Kanehiro, K. Fujiwara, S. Kajita, K. Yokoi, K. Kaneko, H. Hirukawa, Y. Nakamura, and K. Yamane, "Open architecture humanoid robotics platform," in *IEEE Int. Conf. on Robotics and Automation*, 2002, pp. 24–30 vol.1.

[31] H.-M. Maus, S. Revzen, J. Guckenheimer, C. Ludwig, J. Reger, and A. Seyfarth, "Constructing predictive models of human running," *Journal of The Royal Society Interface*, vol. 12, no. 103, 2014.

[32] ——, "Data from: Constructing predictive models of human running. dryad digital repository." 2014. [Online]. Available: http://dx.doi.org/10.5061/dryad.r9v30