Biologically inspired dead-beat controller for bipedal running in 3D

Johannes Englsberger, Pawel Kozlowski, Christian Ott

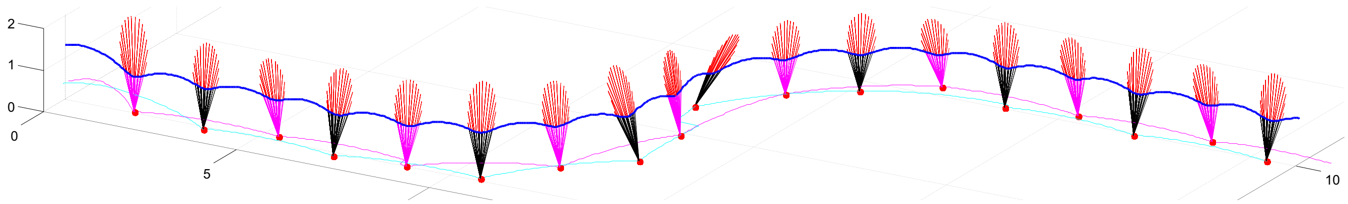

Fig. 1: Bipedal point-mass model runs in 3D based on biologically inspired dead-beat control.

***Abstract*—**
This paper introduces a biologically inspired dead-beat controller for bipedal running in 3D. The controller runs in real-time, is extremely robust against perturbations and allows for very versatile running patterns. It is based on the encoding of leg forces and CoM trajectories during stance as polynomial splines (inspired by observations from human running experiments, see Fig. 2), which allows for an intuitive and primarily analytical controller design. One major advantage of the algorithm is, that both upcoming foot target locations on the ground are predicted at all times, which facilitates the design of appropriate foot trajectories. The performance of the control framework is tested in various simulations for a bipedal point-mass model with two mass-less point feet.

I. 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*. Figure 2 shows a typical GRF profile that was recorded during a human running experiment. The human GRF profiles can be approximated quite well by a polynomial of order 2 in the vertical direction and by a polynomial of order 3 in the $x-$direction. This motivates 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:

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

This linear mapping from the polynomial parameter vectors $\boldsymbol{p}_\sigma$ to CoM positions $\sigma(t)$, velocities $\dot{\sigma}(t)$ and accelerations $\ddot{\sigma}(t)$ was extensively used in the controller derivation.

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

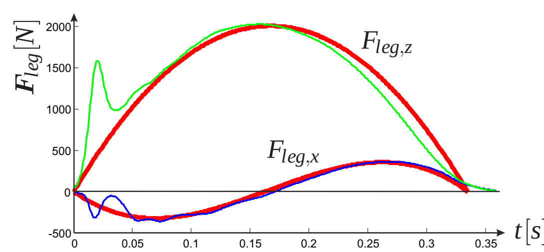

Fig. 2: Comparison of experimentally measured human leg forces (blue/green) and polynomial approximations (red).

II. Derivation of dead-beat controller

Running is typically defined as a locomotion pattern, which employs alternate flight and (single leg supporting) stance phases. In this paper, we use a preview of the upcoming two 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 (at z_{floor}) the apex of the flight curve (i.e. $\dot{z} = 0$) should be and at what CoM height the touch-down is supposed to happen. Another design parameter, used in this work, is the total stance time T_s.

A. Vertical planning and boundary conditions

We make use of four linear vertical boundary conditions:

- initial position equals TD position ($z(t_s = 0) = z_{TD}$)
- initial velocity equals TD velocity ($\dot{z}(t_s = 0) = \dot{z}_{TD}$)
- initial acceleration is minus gravity ($\ddot{z}(t_s = 0) = -g$), i.e. vertical leg force is zero
- final acceleration is minus gravity ($\ddot{z}(t_s = T_s) = -g$), i.e. vertical leg force is zero

These boundary conditions can be solved via a pseudo-inverse. The one additional DOF is used to achieve a desired apex height of the CoM during the upcoming flight phase (analytical solution).

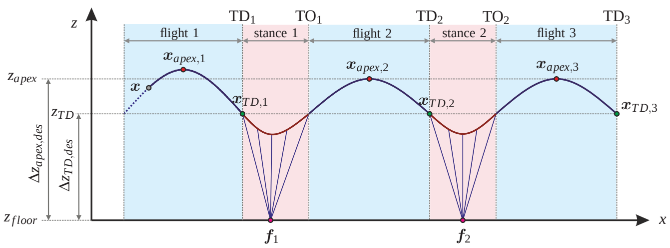

Fig. 3: Preview of upcoming flight and stance phases (planar sketch) - used for design of boundary conditions.

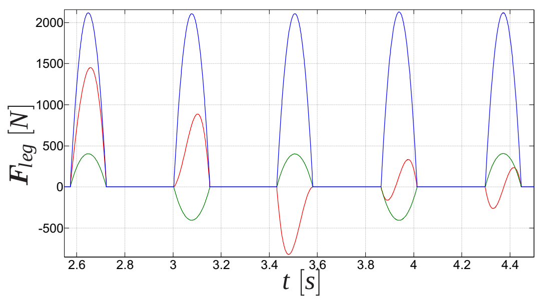

Fig. 4: Force profiles during running simulations.

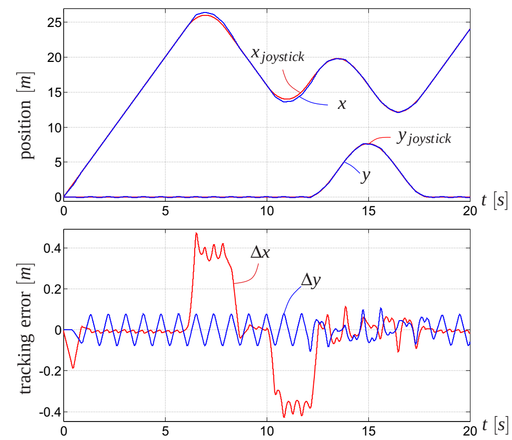

Fig. 5: Tracking performance when steered by joystick input.

B. Horizontal planning and boundary conditions

Motivated by Fig. 2, we the following four linear horizontal boundary conditions ($\chi \in \{x,y\}$):

- initial position equals TD position ($\chi(t_s = 0) = \chi_{TD}$)
- initial velocity equals TD velocity ($\dot{\chi}(t_s = 0) = \dot{\chi}_{TD}$)
- initial acceleration is zero ($\ddot{\chi}(t_s = 0) = 0$), i.e. horizontal leg force is zero
- final acceleration is zero ($\ddot{\chi}(t_s = T_s) = 0$), i.e. horizontal leg force is zero

Additionally, we specify a desired upcoming touch-down position for the CoM, wich can also be expressed as a linear mapping of $\boldsymbol{p}_{\chi}$. Again, these boundary conditions are pre-solved via pseudo-inverse. As compared to the vertical direction, the horizontal directions have one more polynomial parameter - and thus one more degree of freedom (DOF) - available. This DOF, has an effect on the geometry of the leg force rays in space. Our goal is to find the value for $\tilde{p}_{\chi}$, which produces the best possible focusing of leg forces, such that these are best feasible for finite-sized (or even point-) feet. Therefore, the mean square of the deviation (integral over time) of the ground intersection points x_{int} of the leg forces from their mean value $\bar{x}_{int}$ is minimized numerically. The resulting optimal ground intersection points are then used as targets for the foot trajectories.

C. State feedback control

In order to increase the robustness of the control framework, we propose a feed-back control method. It is based on the continuous state-based re-planning of the desired contact forces (and corresponding polynomial parameters) throughout flight phases. This continuous re-planning is facilitated by the very low computational demand of the proposed algorithm. In case of perturbations, we observe very high robustness of the controller (see Sec. III).

III. Simulations

The performance and robustness of the controller were tested in numerous simulations. The model used is a bipedal point-mass robot with point-feet. Figure 4 shows a set of typical force profiles. In the shown example, the robot first accelerates forward, then backward and then switches to a periodic gait. Figure 5 shows our controller's tracking performance for varying joystick inputs. Overall, the controller shows very accurate tracking performance. The higher deviations during some phases are caused by continuously changing joystick inputs, since our steering interface assumes constant steering rates. These deviations might be erased by the use of a different steering interface.

IV. Discussion

The trajectory generation and control method described in this paper yields leg force profiles that are independent of the specific hardware design of some particular robot, i.e. the method is generic. The proposed control framework might be used to identify required actuator characteristics for the design of new robots. On the other side, if a specific robot with its predefined hardware limitations and kinematics is to be controlled, other approaches such as optimal control may be required to assure feasibility and good performance.

Regarding required leg stiffness, we find rather linear stiffness profiles for slow running and more and more nonlinear stiffness profiles for faster running.

In our Matlab/Simulink simulation setup, for a sampling time of 1ms, we were able to compute a bit more than 1000 time steps per second. On a real-time operation system, we expect even shorter computation times.

The algorithm proposed in this paper may also be applied to the problems of hopping and jumping. We also expect that quadrupedal gaits such as galloping and trotting can be achieved by minor adjustments of the algorithm.