

# Integration of vertical COM motion and angular momentum in an extended Capture Point tracking controller for bipedal walking

Johannes Engelsberger, Christian Ott

**Abstract**— In this paper, we demonstrate methods for bipedal walking control based on the Capture Point (CP) methodology. In particular, we introduce a method to intuitively derive a CP reference trajectory from the next three steps and extend the linear inverted pendulum (LIP) based CP tracking controller introduced in [1], generalizing it to a model that contains vertical CoM motions and changes in angular momentum. Respecting the dynamics of general multibody systems, we propose a measurement-based compensation of multi-body effects, which leads to a stable closed-loop dynamics of bipedal walking robots. In addition we propose a ZMP projection method, which prevents the robots feet from tilting and ensures the best feasible CP tracking. The extended CP controller's performance is validated in OpenHRP3 [2] simulations and compared to the controller proposed in [1].

## I. INTRODUCTION

New robotic application areas, such as the exploration of distant planets or the development of modern service robots, increasingly generate the demand on mobile machines, capable of moving over uneven terrain and in the natural human environment. In this context, the ability to step over obstacles as well as the relatively small stance area motivate the use of leg-based robots instead of wheeled systems.

From a control point of view, leg-based systems in particular require the consideration of variable constraints, as well as, depending on the current contact state, the handling of underactuated system equations. To reduce complexity, often simplified models are used, which describe only some essential aspects of the complex multi-body dynamics. The fundamentals of balancing on a limited support area can be investigated using conceptually simple models, which are based on the center of mass (CoM) dynamics and its interplay with the ground reaction forces.

Vukobratovic [3] introduced the Zero Moment Point (ZMP) as an abstract description of the vertical pressure distribution between foot and ground. In spite of their limited validity, simplified models, that correlate the ZMP and the CoM motion, have turned out to be useful for trajectory generation and walking stabilization of several walking machines [4]–[8]. An often used model is the linear inverted pendulum (LIP) model [9], which describes the 'macroscopic' dynamical behavior of bipedal walking

The authors are with Institute of Robotics and Mechatronics, German Aerospace Center, 82234 Weßling, Germany. E-mail: johannes.engelsberger@dlr.de

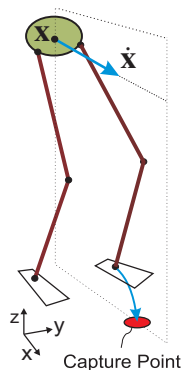


Fig. 1. Illustration of Capture Point



Fig. 2. DLR Biped

machines with constant CoM height astonishingly well. It could be shown that the same model can also be used for walking on slopes and stairs [4].

Based on the LIP model, Kajita et al. [4] proposed an extension of the LQR controller by future reference values for the generation of walking trajectories. This approach was extended to a linear model-predictive control (MPC) by Diedam et. al. [10], which allows an explicit consideration of the ZMP limits as well as an adaption of footprint positions. Although formulated as a feedback control task, these approaches are usually applied for trajectory generation. In addition to the above mentioned pattern generators, Choi et al. [7] presented a proof of stability for a ZMP-based walking stabilizer. In [5] the influence of a non-ideal ZMP dynamics onto the design of a tracking controller was examined. Sugihara [6] considers the limited support area in the parameterization of his stabilizing controller. While often the CoM height is assumed to be constant, a changing CoM height was considered for trajectory generation methods that aimed for ZMP-based running [11]–[13]. In addition to the ZMP-based walking stabilization, the modification of the footstep locations allows for a discrete control of the LIP dynamics [14]–[16].

Based on the system equations of the LIP, the particular ZMP can be calculated that brings the motion of the CoM to a stop. This point was called 'Capture Point' (CP) by Pratt et. al. [17] and 'extrapolated center of mass' by Hof [18]. It is important to notice that in spite of its name "Capture Point", the CP cannot only be used, to bring a robot

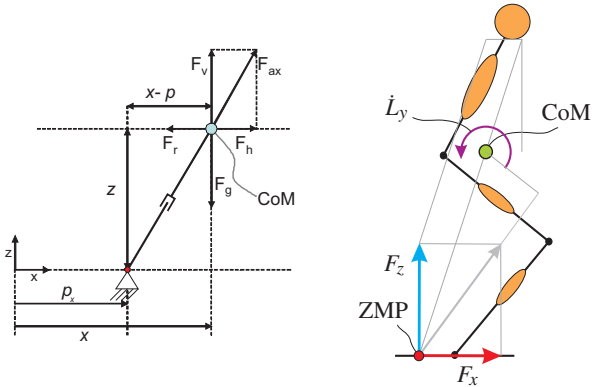


Fig. 3. Robot models discussed in this paper:  
(a) linear inverted pendulum (LIP) (b) Multi-body model

to a stop, but also to control an arbitrary walking motion. By means of the CP, Pratt introduced 'n-step-Capturability Regions'<sup>1</sup>, whereas Hof describes simple step-to-step controllers<sup>2</sup>, which are based on the CP dynamics. Koolen et al. propose a Capture Point based method for walking control, which allows for a footstep adaptation, in [19], [20].

In [21] a real-time walking pattern generation based on an eigenvalue decomposition of the LIP dynamics was proposed, which leads to the definition of a 'converging' and a 'diverging' component. It can be shown that the 'diverging' component from [21] is equivalent to the CP.

In this paper, the CP-based methodology for walking control proposed in [1] is extended to a more general model. We show, that using the CP methodology, the planning of reference trajectories for the CP is very simple and it is sufficient to only stabilize the unstable part of the system dynamics. The previous version of a corresponding CP controller was restricted to walking with constant CoM height. In this paper we show that the CP control can be extended to a nonlinear dynamics, in which the vertical motion of the CoM as well as a varying angular momentum are considered. This allows for better tracking of the CP reference trajectories.

The paper is structured as follows. In section II we introduce a method for finding a desired CP reference trajectory based on the preview of the next three footprints and analyze its convergence. In section III we extend the originally LIP-based CP controller from [1] to a general multi-body model. In section IV we describe a method for projecting the desired ZMP onto the support polygon (convex hull of foot contact area) in order to avoid tilting of the robots feet. Section V describes the position-based ZMP controller used in this work. Section VI gives an overview of the overall control structure and outlines simulation results, which validate the proposed extended CP controller. Section VII summarizes the proposed method and concludes the paper.

<sup>1</sup>regions to which the robot has to step, to come to a stop after  $n$  steps

<sup>2</sup>controllers with only one control action per step

## II. LIP-BASED GENERATION OF CP REFERENCE TRAJECTORIES AND TRACKING CONTROL

### A. Background: basic CP dynamics and control

The first sections in this paper build up on the linear inverted pendulum (LIP) model (Fig. 3(a)) and the Capture Point (CP) [1], [17], [18]. For simplicity, in this section we derive all equations only for the x-direction (y-direction is equivalent and independent). The LIP model uses the assumption, that the robot consists of a point mass (corresponding to the robots center of mass (CoM)) and the CoM is kept at a constant height. The robots ZMP  $p_x$  is equivalent to the torque-free base joint of the LIP. With these assumptions, the horizontal acceleration (see [9]) can be found as

$$\ddot{x} = \omega^2(x - p_x), \quad (1)$$

where  $\omega = \sqrt{g/z}$ . The Capture Point  $\xi$  was derived in [17] and is defined as

$$\xi_x = x + \frac{\dot{x}}{\omega}. \quad (2)$$

Using the CP as state variable, we can write the overall system dynamics in the form

$$\begin{bmatrix} \dot{x} \\ \dot{\xi}_x \end{bmatrix} = \begin{bmatrix} -\omega & \omega \\ 0 & \omega \end{bmatrix} \begin{bmatrix} x \\ \xi_x \end{bmatrix} + \begin{bmatrix} 0 \\ -\omega \end{bmatrix} p_x. \quad (3)$$

We find that the CoM has a stable first order dynamics, whereas the CP has an unstable first order dynamics with the ZMP  $p_x$  as input. Our approach is to stabilize only the unstable CP dynamics, while utilizing the natural stability of the CoM dynamics. Therefore, we derive the solution in time for the CP dynamics assuming a constant ZMP  $p_x$ :

$$\xi_x(T) = e^{\omega T} \xi_x(0) + (1 - e^{\omega T}) p_x, \quad (4)$$

where  $T$  denotes an arbitrary time difference. Solving (4) for  $p_x$  and inserting the current CP  $\xi_x$  as the initial CP leads us to a CP feedback control law

$$p_{x,d} = \underbrace{\frac{1}{1 - e^{\tau}}}_{k_d} \xi_{x,target} - \underbrace{\frac{e^{\tau}}{1 - e^{\tau}}}_{k_d - 1} \xi_x, \quad (5)$$

where  $\tau = \omega T_{ech}$  ( $ech = \text{"end of control horizon"}$ ). Equation (5) gives the desired ZMP  $p_{x,d}$ , which asymptotically controls the CP from the initial CP  $\xi_x$  to a target-CP  $\xi_{x,target}$ . This derivation is primarily based on the assumption that both the target-CP  $\xi_{x,target}$  and the input ZMP  $p_{x,d}$  are constant. However, we use (5) as feedback control law and we will show in section II-C that the control law (5) results in a very convenient behavior of the (nonconstant) CP and ZMP trajectories for the tracking of a (nonconstant) target-CP trajectory. With (5) the closed loop dynamics is

$$\begin{bmatrix} \dot{x} \\ \dot{\xi}_x \end{bmatrix} = \begin{bmatrix} -\omega & \omega \\ 0 & k_d \omega \end{bmatrix} \begin{bmatrix} x \\ \xi_x \end{bmatrix} + \begin{bmatrix} 0 \\ -k_d \omega \end{bmatrix} \xi_{x,target}. \quad (6)$$

As  $k_d < 0$  holds for  $T_{ech} > 0$ , for a constant  $\omega > 0$  the eigenvalues of (6) are stable. In [1], the robustness of (5) w.r.t. CoM modeling errors and ZMP-lag was analyzed.

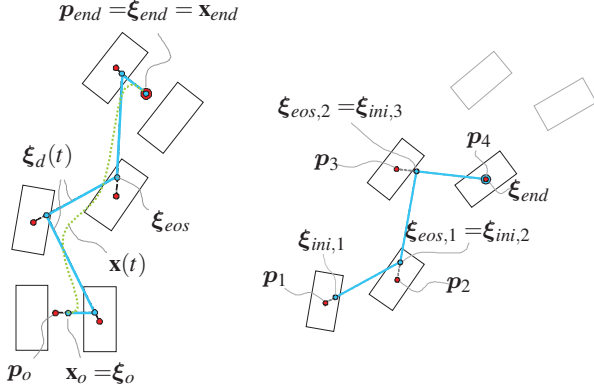


Fig. 4. Two-dimensional Capture Point manipulation: (a) Foot to foot shift (b) Preview of next three steps for CP reference

### B. CP reference trajectory generation

The basics of Capture Point control were introduced in the last section, only considering the  $x$ -direction for simplicity. We will now introduce the two-dimensional CP dynamics, which allows us to control the robots CoM in both horizontal directions ( $x$  and  $y$ ). Extending (4) to 2D leads to

$$\dot{\xi}(t) = \mathbf{p} + e^{\omega t} (\xi_o - \mathbf{p}), \quad (7)$$

where  $\xi = [\xi_x, \xi_y]^T$ ,  $\xi_o = [\xi_{x,o}, \xi_{y,o}]^T$  and  $\mathbf{p} = [p_x, p_y]^T$ . For a constant ZMP  $\mathbf{p}$  the CP  $\xi$  moves away from the ZMP on a straight line. With the two-dimensional CoM dynamics

$$\dot{\mathbf{x}} = -\omega (\mathbf{x} - \xi), \quad (8)$$

we find that the CoM follows the CP, its velocity vector  $\dot{\mathbf{x}}$  always pointing towards the CP.

The basic idea used in this section is to produce a walking pattern by shifting the CP during a step from one footprint to the next (Fig. 4(a)). As the CoM automatically follows the CP (dotted curve), only the CP dynamics (blue lines) has to be considered. This way, the CP and CoM are shifted from the initial CoM position  $\mathbf{x}_o$  to the final CoM position  $\mathbf{x}_{end}$ . The goal CP at the end of each step is denoted by  $\xi_{eos}$  (“eos” = “end of step”).

In this paper we propose the computation of  $\xi_{eos}$  based on a backward calculation (in [1]  $\xi_{eos}$  was chosen heuristically as a hand-tuned offset from the foot center). With the final CP position  $\xi_{eos,i}$  at the end of each step and the desired ZMP position (here chosen to be in the center of the stance foot  $\mathbf{p}_i$ ) we can calculate a desired initial CP  $\xi_{ini,i}$  for each step with  $\xi_{ini,i} = \mathbf{p}_i + (\xi_{eos,i} - \mathbf{p}_i) / (e^{\omega t_{step}})$ , where  $t_{step}$  denotes the total time per step. This desired initial CP  $\xi_{ini,i}$  is then used as the desired final CP position  $\xi_{eos}$  for the previous step, so  $\xi_{eos,i-1} = \xi_{ini,i}$ . In that way, from the final step (after which the robot usually comes to a stop) until the current step, all  $\xi_{eos,i}$  as well as the whole future desired trajectory of the CP (blue lines in Fig. 4(a)) can be calculated. In practice, we limit ourselves to the use of the current footprint ( $\mathbf{p}_1$ )

and the three next footprints ( $\mathbf{p}_2$ ,  $\mathbf{p}_3$  and  $\mathbf{p}_4$ , see Fig. 4(b)) for the calculation of the CP tracking reference, instead of using the whole list of future footprints (if available). We use the assumption  $\xi_{ini,4} = \mathbf{p}_4$ , to start the backward calculation. This reduces the computational effort while the deviation from the trajectory generation using all future footprints is marginal. With (7), we find the reference trajectory

$$\xi_d(t) = e^{\omega t} \xi_{ini,1} + (1 - e^{\omega t}) \mathbf{p}_1. \quad (9)$$

corresponding to the first blue line section ( $\xi_{ini,1}$  to  $\xi_{eos,1}$ ,  $\rightarrow$  first step) in Fig. 4(b). Note that  $\xi_d(t_{step}) = \xi_{eos,1}$ . In the unperturbed case all ZMPs are always located in the foot centers, which decreases the likelihood of tilting.

The proposed method is not restricted to the use of predefined footsteps (such as a predefined list of future steps). The use of a continuously changing set of future steps is content of our current research (not included in this paper).

### C. Convergence analysis for CP tracking

In [1] a Capture Point tracking controller (CPT) was derived, which turned out to be very robust. In the CPT approach, the parameter  $T_{ech}$  in (5) is kept constant whereas the target-CP  $\xi_{target}$  follows the CP reference trajectory  $\xi_d(t)$ . The basic idea is to shift the CP during a time span  $T_{ech}$  to the position where also the reference CP  $\xi_d(t)$  will be located after the time span  $T_{ech}$ . By setting  $\xi_{target}(t) = \xi_d(t + T_{ech})$  we find the target-CP as

$$\xi_{target}(t) = e^{\omega(t+T_{ech})} \xi_{ini,1} + (1 - e^{\omega(t+T_{ech})}) \mathbf{p}_1. \quad (10)$$

Inserting  $\xi_{target}(t)$  and the CP control law (5) into the second row of (3) and solving this time-dependent differential equation, results (for a constant time parameter  $T_{ech}$ ) in the following CP trajectory:

$$\xi(t) = \underbrace{e^{\omega t} \xi_{ini,1} + (1 - e^{\omega t}) \mathbf{p}_1}_{\xi_d(t)} + \underbrace{(\xi_o - \xi_{ini,1}) e^{k_d \omega t}}_{\rightarrow 0}. \quad (11)$$

The first part of the right hand side of (11) is equivalent to the reference CP trajectory  $\xi_d(t)$ , whereas the second part correlates to the error between the initial desired CP  $\xi_{ini,1}$  and the initial real CP  $\xi_o$ . As  $k_d < 0$ , this error term converges to zero and  $\xi$  converges to the desired CP trajectory  $\xi_d(t)$ . The solution of the ZMP has the following form:

$$\mathbf{p}(t) = \mathbf{p}_1 + \underbrace{(k_d - 1)(\xi_{ini,1} - \xi_o) e^{k_d \omega t}}_{\rightarrow 0}. \quad (12)$$

From (11) and (12) it is obvious that the CP error  $\Delta \xi(t) = \xi(t) - \xi_d(t)$  and ZMP error  $\Delta \mathbf{p}(t) = \mathbf{p}(t) - \mathbf{p}_1$  converge to zero (the ZMP  $\mathbf{p}(t)$  converges on a straight line towards  $\mathbf{p}_1$ ). Of course this ideal tracking behavior can only be guaranteed, if there are no modeling errors, there is no lag in the generated ZMP and the physical limitations, such as support polygon (see section IV), are not violated.

### III. EXTENDED CP CONTROL, ACCOUNTING FOR VARIATIONS IN CoM HEIGHT AND ANGULAR MOMENTUM

As we saw in the last sections, the dynamics of a LIP can be solved analytically which simplifies the design of walking algorithms for bipedal robots. Though, as robots are usually affected by multi-body effects, for example produced by the swing leg motion, it is often tried [4], [22]–[24] to find ZMP controllers that compensate for such multi-body effects. The 3-mass-model [22] in combination with preplanned foot trajectories is one possible attempt. Usually, the goal is to track the precomputed ZMP reference trajectory more exactly. In this section, we introduce an extension of the CP control for measurement-based compensation of multi-body effects, such as variations in CoM height and angular momentum.

In the following section, we will limit ourselves to the sagittal plane ( $x$ - $z$ -plane), as the dynamics in the  $y$ - $z$ -plane is equivalent and independent. The variables in this section may be time-varying, which is omitted in some cases for better readability. Considering the full multi-body dynamics, we can write the horizontal CoM acceleration as

$$\ddot{x} = \frac{g + \ddot{z}}{z} (x - p_x) + \frac{\dot{L}_y}{m z}, \quad (13)$$

where  $\dot{L}_y$  denotes the change in angular momentum. Controlling a robots ZMP, assuming it behaves like a LIP ( $\rightarrow p_{x,d} = x - \ddot{x} z/g$ , see (1)), leads to the following falsified horizontal CoM acceleration:

$$\ddot{x} = (1 + \frac{\ddot{z}}{g}) \ddot{x}_d + \frac{\dot{L}_y}{m z}, \quad (14)$$

where  $\ddot{x}_d$  denotes the desired and  $\ddot{x}$  the achieved horizontal acceleration. This motivates the consideration of the multi-body effects in the design of feedback controllers for bipedal walking machines.

The definition of the CP  $\xi$  (2) was originally motivated by the LIP dynamics. Nevertheless, we use the state variable  $\xi_x = x + \dot{x}/\omega(t)$  also for the control of the more general model (13). The originally constant  $\omega$  is now a function of time:  $\omega(t) = \sqrt{g/z(t)}$ . Therefore, the CP dynamics is now

$$\dot{\xi}_x = \dot{x} + \frac{\omega(t) \ddot{x} - \dot{\omega}(t) \dot{x}}{\omega(t)^2}. \quad (15)$$

Using (2), (13) and  $\dot{\omega}(t) = -\frac{\omega(t)\dot{z}}{2z}$  the overall system dynamics can be written as

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{\xi}_x \end{bmatrix} &= \begin{bmatrix} -\omega(t) & \omega(t) \\ \frac{\dot{z}}{\omega(t)z} - \frac{\dot{z}}{2z} & \omega(t) + \frac{\dot{z}}{2z} \end{bmatrix} \begin{bmatrix} x \\ \xi_x \end{bmatrix} + \\ &+ \begin{bmatrix} 0 \\ -\frac{g+\ddot{z}}{\omega(t)z} \end{bmatrix} p_x + \begin{bmatrix} 0 \\ \frac{\dot{L}_y}{m\omega(t)z} \end{bmatrix}. \end{aligned} \quad (16)$$

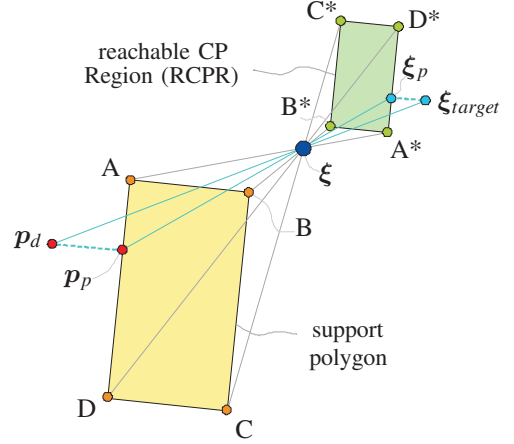


Fig. 5. Support polygon projection of the desired ZMP  $p_d$

To transform this system into a cascaded form, we use the control law

$$p_{x,d} = \frac{g \hat{p}_{x,d} + (\ddot{z} - \frac{\omega(t)\dot{z}}{2}) x + \frac{\omega(t)\dot{z}}{2} \xi_x + \frac{\dot{L}_y}{m}}{g + \ddot{z}}, \quad (17)$$

with an intermediate control input  $\hat{p}_{x,d}$ . For stabilizing the time-varying system we make use of the CP control law (5)

$$\hat{p}_{x,d} = \frac{1}{1 - e^{-\hat{\tau}(t)}} \xi_{x,target} - \frac{e^{-\hat{\tau}(t)}}{1 - e^{-\hat{\tau}(t)}} \xi_x, \quad (18)$$

where  $\hat{\tau}(t) := \omega(t)T_{ech}$  is now time-varying. The use of (17)-(18) results in the following closed-loop dynamics

$$\begin{bmatrix} \dot{x} \\ \dot{\xi}_x \end{bmatrix} = \begin{bmatrix} -\omega(t) & \omega(t) \\ 0 & k_d(t) \omega(t) \end{bmatrix} \begin{bmatrix} x \\ \xi_x \end{bmatrix} + \begin{bmatrix} 0 \\ -k_d(t) \omega(t) \end{bmatrix} \xi_{x,target}. \quad (19)$$

In contrast to (6) the closed-loop dynamics is time-varying. Its stability is proven in the appendix by a Lyapunov analysis.

In the presented work, instead of a precompensation of beforehand known multi-body influences (e.g. 3-mass-model [22]), we propose measurement-based estimation and compensation of beforehand unknown multi-body influences. Therefore, we estimate the vertical acceleration  $\hat{\ddot{z}} = F_z/m - g$  and the change in angular momentum  $\hat{\dot{L}}_y = F_x z - F_z(x - p_x)$  (see [23]), where  $F = [F_x, F_z]$  denotes the measured ground reaction force and  $p_x$  is the measured ZMP. Inserting these estimates into the control law (17)-(18) returns the desired ZMP  $p_{x,d}$ , which results in the stable closed-loop CP dynamics (19).

It has to be noted, that although in the LIP based planning we assume constant ZMP positions (for the unperturbed case), the control law (17)-(18) produces non-constant desired ZMPs, which are not always located within the support polygon. To ensure feasibility, we propose a projection method in the next section.

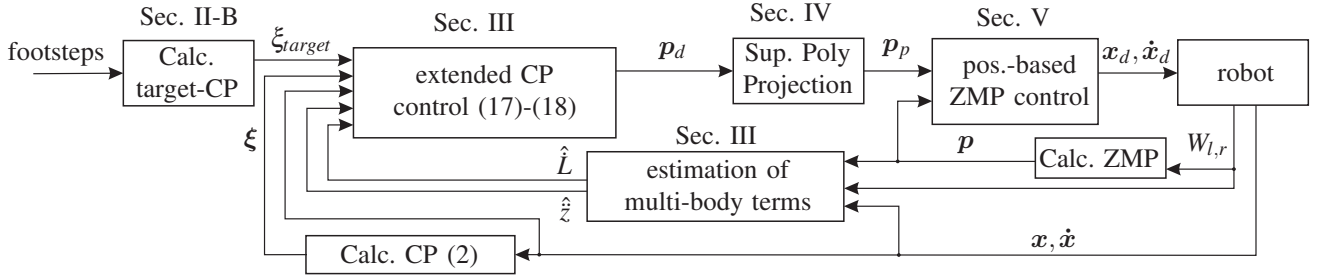


Fig. 6. Overall algorithm for walking control of DLR Biped

#### IV. TILTING AVOIDANCE

The desired ZMP  $p_d = [p_{x,d}, p_{y,d}]$ , produced by the CP control, is not necessarily located within the support polygon. To avoid tilting around an edge of the support polygon,  $p_d$  is projected onto the support polygon. We choose a projection method, which minimizes the Euclidean distance between  $p_d$  and the projected ZMP  $p_p$  (see Fig. 5). We use this projection method deliberately, as it can be shown that it leads to the least possible deviation (in the Euclidean sense) of the real CP from the target-CP trajectory  $\xi_{target}(t)$ . As shown in Fig. 5, for each point on the support polygon (e.g. A) a corresponding final CP (A\*) can be found via a projection over the current CP  $\xi$  (assuming  $p = const$ ):

$$\xi(T_{ech}) = \xi + (e^{\omega T_{ech}} - 1)(\xi - p). \quad (20)$$

This corresponding final CP  $\xi(T_{ech})$  will be reached by the real CP after the time span  $T_{ech}$ , if the according ZMP is chosen. Consequently, via the described forward projection we can calculate the corresponding reachable CP region (RCPR, see Fig. 5), which is reachable within the time span  $T_{ech}$ . As this projection over the current CP  $\xi$  is a linear mapping (scaled point reflection w.r.t. the CP  $\xi$ ), all aspect ratios within the two regions stay equal. It can be shown that the reflection of a vector is equivalent to a dilation by a negative constant factor. Therefore, the minimization of the CP error (Euclidean projection of  $\xi_{target}$  onto the RCPR,  $\rightarrow \xi_p$ ) is equivalent to the minimization of the ZMP error (Euclidean projection of  $p_d$  onto the support polygon,  $\rightarrow p_p$ ). The possibility of explicitly avoiding tilting of the feet by a direct limitation of the ZMP to the support polygon is an advantage compared to other control approaches [4], [7].

It has to be pointed out, that for strong perturbations (more precisely: when the CP leaves the support polygon) the proposed ZMP stabilizer is no longer sufficient to stop the CoM from diverging. In such cases, other methods, such as an appropriate foot placement (e.g. described by Stephens et al. in [25]) or the direct use of a torque around the CoM for stabilizing the robot, have to be used additionally.

#### V. POSITION BASED ZMP CONTROL

To produce the desired ZMP from (17) on a position controlled robot, an underlying ZMP control loop is required (shown for  $x$ -direction). Using (13), we relate the desired ZMP  $p_{x,d}$  to a desired force  $F_{x,d} = m \ddot{x}_d = (x - p_{x,d}) F_z/z + \dot{L}_y/z$ , which should act on the CoM. A desired force can be generated by a proportional controller of the form [26]:

$$\ddot{x}_d = k_f(F_{x,d} - F_x), \quad (21)$$

with  $k_f > 0$ . Since we are interested in a ZMP controller rather than a force controller, we insert  $F_{x,d}$  and the equivalent relation between the force  $F_x$  and  $p_x$  into (21) to obtain a desired change in CoM velocity

$$\dot{x}_d = k_f F_z/z (p_x - p_{x,d}). \quad (22)$$

The desired CoM position and velocity  $x_d$  and  $\dot{x}_d$  result from integration of (22) and are commanded to the robot ( $y$ -direction equivalent).

#### VI. OVERALL STRUCTURE OF THE CP CONTROL AND SIMULATION RESULTS

Figure 6 gives an overview of the overall structure of the proposed control algorithm. From a set of future footprints, a target-CP reference  $\xi_{target}(t)$  is calculated (section II-B). The CP is calculated from the current robot state  $(x, \dot{x})$  and the ZMP  $p$  from the wrenches  $W_{l,r}$  in the left and right foot.  $\hat{L}$  and  $\hat{z}$  are estimated and used in the extended CP controller (section III), which returns a desired ZMP  $p_d$ . The desired ZMP is (if necessary) projected onto the support polygon (SP, section IV) to avoid tilting of the robot. The projected ZMP  $p_p$  is then transferred to a ZMP controller, which receives the measured ZMP  $p$  as feedback, and commands a desired CoM position and velocity to the robot.

The support state (single/double support) is detected online over force-torque sensors in the robots feet and the SP is calculated online, knowing the foot geometry. Therefore, the control can handle both single and double support states (current SP is used for ZMP projection) although the planning is based only on single support phases.

To validate the performance of the proposed extended CP control, simulations in OpenHRP3 [2] were carried out.

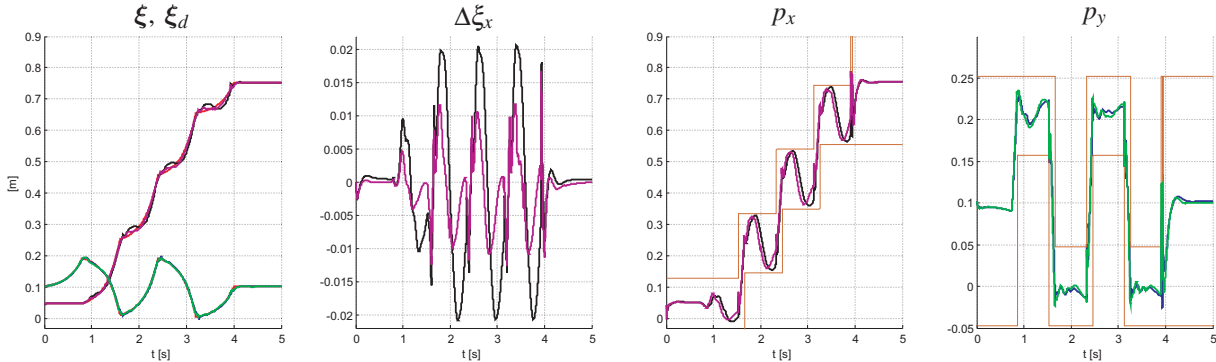


Fig. 7. Open HRP Simulation of LIP-based CP control (5) and extended CP control (section III) for a straight walk ( $l_{step} = 20\text{cm}$ ,  $t_{step} = 0.8\text{s}$ ), LIP-based CP control:  $x \rightarrow$  black,  $y \rightarrow$  blue; extended CP control:  $x \rightarrow$  pink,  $y \rightarrow$  green; CP reference: red; limits of support polygon: brown

Figure 7 shows a simulation of a straight walk (step length 20cm,  $t_{step}$  0.8s) during which the DLR Biped (see fig. 2) performed four steps and then came to a rest. Here, the LIP-based CP controller (5) (proposed in [1]) and the extended CP controller (III), proposed in this paper, were compared. Both algorithms show a quite good tracking of the desired CP trajectory (red curve). The tracking in  $x$ -direction shows to be more perturbed than the  $y$ -direction, which is caused by the acceleration of the feet in the sagittal plane ( $x$ - $z$ -plane) and the corresponding higher deviations from the LIP dynamics. We find that the proposed extended CP controller shows smaller deviations from the CP tracking reference than the LIP-based controller. The second graph shows the CP tracking error in  $x$ -direction. The mean absolute tracking error of the CP in  $x$ -direction was 7.3mm for the LIP-based CP controller (5) and 3.2mm for the proposed extended CP controller (17)-(18), which motivates the use of measurement-based compensation of multi-body effects. For both controllers the ZMPs stay within the support polygon (brown line) during the whole simulation.

The choice of a straight walk for the simulation was made to keep the shown trajectories and directional dependencies (e.g. tracking in  $x$  and  $y$  direction, Fig. 7) as simple as possible. The proposed control algorithm can manage walks including both arbitrary step positions and orientations, without any increase of complexity compared to a straight walk.

## VII. SUMMARY AND CONCLUSIONS

The Capture Point methodology describes the dynamics of a linear inverted pendulum (LIP) as two cascaded first order systems, by what the system dynamics becomes very intuitively comprehensible. The ZMP  $p$  is used as the input variable of the system, which stabilizes the robot and allows for a direct manipulation of the CP. Through the intrinsic stability of the CoM dynamics the CP methodology allows for an effective and robust control of bipedal walking machines. In this paper, we introduced a method to intuitively derive a CP reference trajectory from the next three steps

for the CP tracking controller proposed in [1]. We extended our control to the measurement-based compensation of 3D-effects (vertical CoM motion and varying angular momentum of the robot), which allows for more accurate tracking of the desired CP reference. Although the original control laws proposed in [1] perform well for standard walking cases, we could show the even better performance of the extended control laws proposed in this paper. The extension considers the whole multi-body dynamics of the robot, which can be seen as a clearly more thorough theoretical treatment compared to the LIP model. Therefore, for more advanced bipedal locomotion we expect a way better performance of the extended control method compared to LIP based ones.

In addition to the extended CP control laws, we proposed a ZMP projection method, which prevents the robot from tilting and ensures the best feasible CP tracking. The robustness of the proposed control strategy was validated in simulations using OpenHRP3 [2]. All computations for the proposed CP control algorithm can be performed online (at a 1ms sampling time in our system). Compared to model predictive control [8] the far lower computation effort of CP control proves to be advantageous. Furthermore, preview controllers and model predictive controllers are usually primarily used for trajectory generation with subsequent stabilizing controller, whereas the CP methodology is a self-contained solution for CP reference generation (feed-forward) and CP tracking control (feed-back). Another advantage of CP control over some other concepts [4], [7] is, that it allows for an explicit limitation of the ZMP to the support polygon, which decreases the danger of tilting.

Based on the presented methods, we intend to develop methods for (i) online footprint adjustment and (ii) force controlled walking (based on [27]).

## ACKNOWLEDGEMENTS

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).

APPENDIX

*Lemma 1: Given a linear time-varying second order system in cascaded form*

$$\dot{x}_1 = -a(t)x_1 + a(t)x_2 \quad (23)$$

$$\dot{x}_2 = -b(t)x_2 \quad (24)$$

with the state variables  $x_1 \in \mathbb{R}$  and  $x_2 \in \mathbb{R}$ . If the scalar functions  $a(t)$  and  $b(t)$  are strictly positive, i.e.  $\exists a_m > 0$ ,  $b_m > 0$ ,  $\forall t$ ,  $a(t) > a_m$ ,  $b(t) > b_m$ , and  $a(t)$  is bounded above, i.e.  $\forall t$ ,  $a(t) < a_M$ , the system is exponentially stable.

*Proof:* Since the system is time-varying the stability proof cannot be based on linear analysis. Instead, we utilize the Lyapunov function  $V(x_1, x_2) = x_1^2/2 + \varepsilon x_2^2/2$ , with  $\varepsilon > a_M^2/(4\delta^2 a_m b_m)$  and  $0 < \delta < 1$ . The derivative of the Lyapunov function along the systems solutions is given by

$$\begin{aligned} \dot{V}(x_1, x_2) &= -a(t)x_1^2 + a(t)x_1x_2 - \varepsilon b(t)x_2^2 \\ &= -\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}^T (\mathbf{M}_1 + \mathbf{M}_2) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ \mathbf{M}_1 &= \begin{bmatrix} a_m(1-\delta) & 0 \\ 0 & \varepsilon b_m(1-\delta) \end{bmatrix} \\ \mathbf{M}_2 &= \begin{bmatrix} a(t) - a_m(1-\delta) & -\frac{1}{2}a(t) \\ -\frac{1}{2}a(t) & \varepsilon(b(t) - b_m(1-\delta)) \end{bmatrix} \end{aligned}$$

Since matrix  $\mathbf{M}_1$  is a constant positive definite term, we only need to show, that also matrix  $\mathbf{M}_2$  stays positive definite for concluding exponential stability.  $\mathbf{M}_2$  is positive definite, if the two conditions  $a(t) - a_m(1-\delta) > 0$  and  $\varepsilon > \frac{1}{4} \frac{a(t)^2}{(a(t) - a_m(1-\delta))(b(t) - b_m(1-\delta))}$  hold (see [28]). The former condition is fulfilled since  $a(t) > a_m$  is strictly positive. The latter condition is ensured by the choice  $\varepsilon > \frac{1}{4} \frac{a_M^2}{\delta^2 a_m b_m}$ . ■

REFERENCES

[1] J. Engelsberger, C. Ott, M. A. Roa, A. Albu-Schäffer, and G. Hirzinger, "Bipedal walking control based on capture point dynamics," in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2011, pp. 4420–4427.

[2] 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.

[3] M. Vukobratovic and Y. Stepanenko, "On the stability of anthropomorphic systems," *Mathematical Biosciences*, vol. 15, pp. 1–37, 1972.

[4] 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.

[5] 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 *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2010, pp. 4489–4496.

[6] T. Sugihara and Y. Nakamura, "Boundary condition relaxation method for stepwise pedipulation planning of biped robots," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 658–669, 2009.

[7] Y. Choi, D. Kim, Y. Oh, and B.-J. You, "Posture/walking control for humanoid robot based on kinematic resolution of com jacobian with embedded motion," *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1285–1293, 2007.

[8] P.-B. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *IEEE-RAS International Conference on Humanoid Robots*, 2006.

[9] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation," in *IEEE Int. Conf. on Robotics and Automation*, 2001, pp. 239–246.

[10] H. Diedam, D. Dimitrov, P.-B. Wieber, K. Mombaur, and M. Diehl, "Online walking gait generation with adaptive foot positioning through linear model predictive control," in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2008, pp. 1121–1126.

[11] S. Kajita, K. Kaneko, M. Morisawa, S. Nakaoka, and H. Hirukawa, "Zmp-based biped running enhanced by toe springs," in *IEEE Int. Conf. on Robotics and Automation*, 2007, pp. 3963–3969.

[12] T. Takenaka, T. Matsumoto, T. Yoshiike, and S. Shirokura, "Real time motion generation and control for biped robot, 2nd report: Running gait pattern generation," in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2009, pp. 1092–1099.

[13] R. Tajima, D. Honda, and K. Suga, "Fast running experiments involving a humanoid robot," in *IEEE Int. Conf. on Robotics and Automation*, 2009, pp. 1571–1576.

[14] M. A. Townsend, "Biped gait stabilization via foot placement," *Journal of Biomechanics*, vol. 18, pp. 21–38, 1985.

[15] M. S. Redfern and T. Schumann, "A model of foot placement during gait," *Journal of Biomechanics*, vol. 27, pp. 1339–1346, 1994.

[16] J. Hofschulte, "Closed loop controlled foot placing for bipedal walking robots," in *CISM-IFTOMM Symposium on Robot Design, Dynamics, and Control (ROMANSY)*, 2008, pp. 417–424.

[17] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 200–207.

[18] A. L. Hof, "The 'extrapolated center of mass' concept suggests a simple control of balance in walking," *Human Movement Science*, vol. 27, pp. 112–125, 2008.

[19] 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," *IJRR*, 2012.

[20] J. E. Pratt, T. Koolen, T. D. Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson, and P. Neuhaus, "Capturability-based analysis and control of legged locomotion, part 2: Application to m2v2, a lower-body humanoid," *IJRR*, 2012.

[21] T. Takenaka, T. Matsumoto, and T. Yoshiike, "Real time motion generation and control for biped robot, 1st report: Walking gait pattern generation," in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2009, pp. 1084–1091.

[22] T. Buschmann, S. Lohmeier, M. Bachmayer, H. Ulbrich, and F. Pfeiffer, "A collocation method for real-time walking pattern generation," in *HUMANOIDS*, 2007.

[23] R. Beranek, H. Fung, and M. Ahmadi, "A walking stability controller with disturbance rejection based on cmp criterion and ground reaction force feedback," in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2011, pp. 2261–2266.

[24] T. Takenaka, T. Matsumoto, and T. Yoshiike, "Real time motion generation and control for biped robot, 3rd report: Dynamics error compensation," in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2009, pp. 1594–1600.

[25] B. Stephens and C. Atkeson, "Push recovery by stepping for humanoid robots with force controlled joints," in *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, 2010, pp. 52–59.

[26] J. Roy and L. Whitcomb, "Adaptive force control of position/velocity controlled robots: Theory and experiment," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, pp. 121–137, 2002.

[27] C. Ott, M. A. Roa, and G. Hirzinger, "Posture and balance control for biped robots based on contact force optimization," in *IEEE-RAS International Conference on Humanoid Robots*, 2011, pp. 26–33.

[28] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1990.