

Omnidirectional Capture Steps for Bipedal Walking

Marcell Missura and Sven Behnke

Abstract—Robust walking on two legs has proven to be one of the most difficult challenges of humanoid robotics. Bipedal walkers are inherently unstable systems that are difficult to control due to the complexity of their full-body dynamics. Aside from the challenge of generating a walking motion itself, closed-loop algorithms are required to maintain the balance of the robot using foot placements and other disturbance-rejection strategies.

In this work, we propose a hierarchical, omnidirectional gait control framework that is able to counteract strong perturbations using a combination of step-timing, foot-placement, and zero-moment-point strategies. The perturbations can occur from any direction at any time during the step. The controller will not only maintain balance, but also follow a given reference locomotion velocity while absorbing the disturbance. The calculation of the timing, the footstep locations, and the zero moment point is based on the linear inverted pendulum model and can be computed efficiently in closed form.

I. INTRODUCTION

Replicating the efficiency, stability, and grace of the natural human gait is one of the major challenges of humanoid robotics. Bipedal walkers are inherently unstable and difficult to control. The ground projection of their center of mass (CoM) lies outside of their support polygon most of the time during the gait cycle and, thus, static stability is not present. As a consequence, continuous action is required to prevent the system from falling to the ground, such as adequate motion of the support leg and the torso to keep the zero moment point (ZMP) within the boundaries of the support foot, and placement of the next support at the right time and at the right place. To absorb sudden disturbances that would destabilize the walker, a quick and appropriate reaction is

All authors are with the Autonomous Intelligent Systems Group, University of Bonn, Germany. Email: missura@ais.uni-bonn.de. This work has been partially supported by grant BE 2556/6-1 of German Research Foundation (DFG).

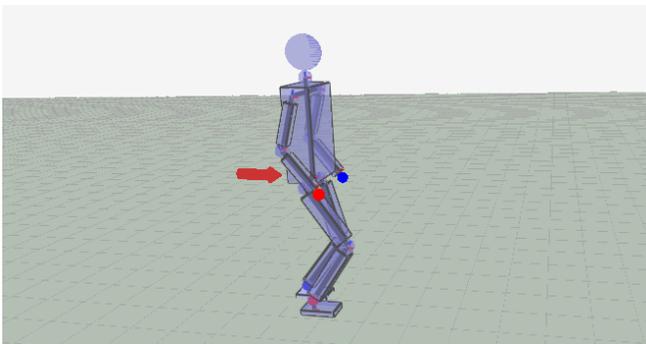


Fig. 1. Our simulated robot recovers from pushes in any direction.

essential. Unfortunately, the computation of dynamic full-body motions for a high number of degrees of freedom is a difficult task. The application on real hardware imposes additional challenges on the design of control algorithms, such as low computational power of embedded systems, noisy sensors, friction, backlash, and latency in the entire sensorimotor control loop.

In this work, we propose a new, closed-loop control approach with a hierarchical layout. We decouple the tasks of balance control and motion generation and combine them in a layered framework. The balance controller computes the timing and the coordinates of the next footstep, and a desired ZMP location. These step parameters are sufficient inputs for the generation of step motions in the bottom layer. Any existing motion trajectory generator can in principle be used for this purpose.

The balance controller is characterized by a strong distinction between the lateral and the sagittal dynamics. The step parameters are computed analytically using very little computational power. The balance controller is able to absorb disturbances that can occur from any direction and at any time during a step. The algorithm will aim for a one step recovery, but due to deviation of the full-body dynamics from the strongly simplified point mass model, kinematic limitations, latency, and imperfect actuation, the effect of a capture step can be reduced. Residual instability will, however, be absorbed during the following step. The system automatically takes as many steps as it needs to return to a nominal walk cycle that reflects the desired walking velocity.

The requirements on hardware and software components for our algorithm to work are very low. Required sensors are joint angle sensors and an inertial measurement unit to estimate the torso attitude. Foot pressure sensors can be helpful for detecting ground contact, but they are not essential. Furthermore, a kinematic model of the robot is used to estimate the state of the CoM, that we approximate with a fixed point on the upper body in the center between the hip joints. For this purpose, forward kinematic computation is sufficient. A dynamic model of the robot, i.e. the masses and the mass distribution of body parts, are not needed. While our algorithm expresses a desired ZMP offset, we never actually measure the true ZMP location. The step motion generator we implemented for our simulation makes use of inverse kinematics. Alternatively, an existing walking motion generator could be used that relies only on forward kinematics [1], [2] in order to eliminate the inverse kinematics component.

II. RELATED WORK

ZMP tracking with preview control is the most popular approach to bipedal walking to date. ASIMO [3], HRP-4C [4], and HUBO [5] are among the most prominent examples. A number of footsteps planned ahead in time that define a ZMP trajectory are used as the reference input. Optimization algorithms are generally used in a Model Predictive Control [6] setting to generate a continuous CoM trajectory that minimizes the ZMP tracking error. The CoM trajectory can then be used in combination with inverse kinematics and high-gain position control, or inverse dynamics torque control to generate joint motions. These systems can walk reliably on flat ground and have the ability to cope with weak disturbances. However, their nature of following a sequence of predefined steps with a fixed frequency prevents a flexible response to strong perturbations that require a quick change of footstep locations and timing in order to maintain balance.

Our method does not constrain the location of the next footstep. On the contrary, the ZMP and the next footstep location are computed based on the current state of the center of mass. Using a CoM reference trajectory instead of a ZMP reference trajectory has the benefit that ZMP and footstep locations arise naturally without the need for additional computation, while the inverse problem of finding a suitable CoM trajectory that satisfies a given ZMP trajectory is difficult to solve and can be computationally expensive.

To recover from strong pushes and collisions, reactive stepping is necessary, as demonstrated by the amazing performance of the quadruped BigDog [7]. Among bipedal walkers, reactive stepping is a new discipline. Toyota's running robot [8] and HUBO [9] demonstrated the ability to cope with a frontal push against the chest during hopping on the spot. A popular approach is to combine momentum suppression and reactive stepping [9], [10], [11] so that a step is only taken if the disturbance cannot be compensated otherwise. From these proposals only the approach of Morisawa et al. [10] is able to react to a push from any direction. Their method also requires an estimation of external forces, and it does not attempt to adjust step timing.

Urata et al. have presented an impressive foot-placement based controller on a real robot that is capable of recovering from strong pushes [12]. To speed up execution time, an LQ preview based algorithm is used to generate ZMP trajectories that can be tracked by the CoM without the need for further optimization. The algorithm generates a recovery sequence of a fixed number of steps that are found by sampling a set of feasible ZMP trajectories and uses a full-body dynamics model for accurate tracking.

Englsberger et al. [13] presented a gait pattern generator based on capture point regulation and showed how using the capture point as the input instead of the ZMP reduces the system equations to first order. However, only ZMP control and no foot placement was considered. We have adopted parts of this work in our framework to implement ZMP regulation that does not require measuring of the actual ZMP.

Extensive work on stability analysis of bipedal systems

has been presented by Pratt et al. [14] based on capture point dynamics. The capture point [15] is the location on the ground where a biped needs to step in order to come to a stop. A comprehensive and analytically tractable formalism was introduced to compute regions of N-step capturability for simple bipedal models that include a support area of non-zero size and a hip torque driven reaction mass. Based on conclusions from the analysis, a bipedal gait controller that aims footsteps at the 1-step capture region was successfully implemented in simulation and on a real robot. Adaptation of step timing was not considered and disturbances with a direction towards the current support leg have been excluded. A limitation to the 1-step capture region may be too restrictive, as the number of steps needed to stop increases with walking velocity. Furthermore, while the capture point is an excellent indicator of stability, perhaps it is not the best concept to find suitable step locations, as a walker typically does not want to stop, but maintain a reference speed, even after a strong disturbance like tripping or a push. Our approach to find appropriate step sizes is based on the mapping of the measured CoM state to a limit cycle that would result in the same CoM velocity, and deducing the step size that would produce that limit cycle.

In previous work [16] we proposed a lateral capture step controller that was implemented in this framework and evaluated on a real robot that was able to recover from strong pushes in lateral direction during walking. Pushes were allowed to tip the robot even in the direction of the support leg and the support foot was not assumed to remain flat on the ground at all times. Also, a combination of step timing and an angular momentum regulator was successfully implemented on a Nao robot [17].

III. MOTIVATION

The pendulum-like dynamics of human walking has been long known to be a principle of energy-efficient locomotion [18]. Figure 2 shows stick diagrams of the idealized sagittal and lateral pendulum motions projected on the sagittal plane and the frontal plane. Interestingly, the sagittal and lateral motions exhibit opposing behaviors. In the sagittal plane, the center of mass "vaults" over the pivot point in every gait cycle while in the frontal plane the center of mass oscillates between the support feet and never crosses the pendulum pivot point.

It is crucial not to tip over sideways, as the recovery from such an unstable state requires challenging motions

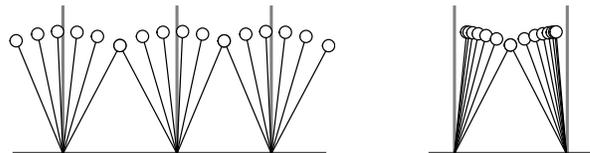


Fig. 2. Stick diagrams of the idealized pendulum-like sagittal motion (left) and lateral motion (right) of a compass gait. In sagittal direction, the center of mass crosses the pendulum pivot point in every gait cycle while in lateral direction it is crucial that the pendulum never crosses the pivot point.

that humanoid robots have difficulties performing. The small lateral distance between the pivot point and the center of mass at the apex of the step provides only a narrow margin for error and constrains the lateral coordinate of the next footstep. Furthermore, the perpetual lateral oscillation of the center of mass, a consequence of the absence of static stability, appears to be the primary determinant of the step timing. Disobeying the right timing can quickly destabilize the system [19]. It is not surprising that substantial effort needs to be invested in lateral control [20] [21].

In the sagittal direction, however, the situation is entirely different. Following from the law $v = \phi l$, any walking velocity v can be the result of an infinite amount of step frequency ϕ and stride length l combinations. Therefore, in sagittal direction, the biped can flexibly accommodate variations in timing with a change of the stride length, e.g. take a short and quick step or a long and slow one, and still maintain a constant walking velocity.

Consequently, we formulate the following control principles for our balance control computations:

- The timing of the steps is determined by the lateral direction when the CoM reaches a nominal support exchange location roughly in the center of the stride width.
- The lateral step size is chosen so that the CoM will pass the following step apex with a nominal distance to the pivot point.
- Variations in timing are accommodated by the choice of the sagittal stride length while maintaining a desired walking velocity.

IV. THE LINEAR INVERTED PENDULUM MODEL

The linear inverted pendulum is a simple mathematical model that serves as an approximation of the principle dynamics of human walking [22]. It describes a motion in one dimension governed by the equation

$$\ddot{x} = C^2 x, \quad (1)$$

where C is a constant typically chosen to be $C = \sqrt{\frac{g}{h_{CoM}}}$, with the gravitational constant $g = 9.81 \frac{m}{s^2}$ and h_{CoM} the height of the center of mass. Given an initial state (x_0, \dot{x}_0) , the equations

$$x(\Delta t) = x_0 \cosh(C\Delta t) + \frac{\dot{x}_0}{C} \sinh(C\Delta t) \quad (2)$$

$$\dot{x}(\Delta t) = x_0 C \sinh(C\Delta t) + \dot{x}_0 \cosh(C\Delta t) \quad (3)$$

compute the future state at time Δt , and

$$t(x) = \frac{1}{C} \ln \left(\frac{x}{c_1} \pm \sqrt{\frac{x^2}{c_1^2} - \frac{c_2}{c_1}} \right) \quad (4)$$

$$t(\dot{x}) = \frac{1}{C} \ln \left(\frac{\dot{x}}{c_1 C} \pm \sqrt{\frac{\dot{x}^2}{c_1^2 C^2} + \frac{c_2}{c_1}} \right) \quad (5)$$

compute the time t when the center of mass reaches a future location x or velocity \dot{x} , using

$$c_1 = x_0 + \frac{\dot{x}_0}{C}, \quad (6)$$

$$c_2 = x_0 - \frac{\dot{x}_0}{C}. \quad (7)$$

Furthermore, unless the pendulum is disturbed by external forces, the orbital energy

$$E = \frac{1}{2}(\dot{x}^2 - C^2 x^2) \quad (8)$$

is constant for an entire trajectory.

To model sagittal and lateral motion, two uncoupled pendulum equations are used:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} C^2 & 0 \\ 0 & C^2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \quad (9)$$

V. HIERARCHICAL FRAMEWORK

To simplify the modeling and control of a closed-loop bipedal gait, we use a hierarchical structure to separate balance control from motion generation, as illustrated in Figure 3. The input into the entire system is a velocity vector $V = (V_x, V_y, V_\theta)$ that expresses the reference walking velocity in sagittal, lateral, and rotational directions. Based only on the desired walking velocity, a nominal CoM state s is computed that the Balance Control layer will attempt to reach in the moment of the next support exchange. Inside the Balance Control layer, ZMP and step parameters are computed in a sequential order that resolves dependencies. The resulting step parameters are passed on to the motion generation layer that outputs joint angle targets, which are either executed in a physical simulator or on a real robot. The loop is closed by using the joint angles and the torso attitude as measured from the robot to compute the posture of a kinematic model, from which the sign of the foot that

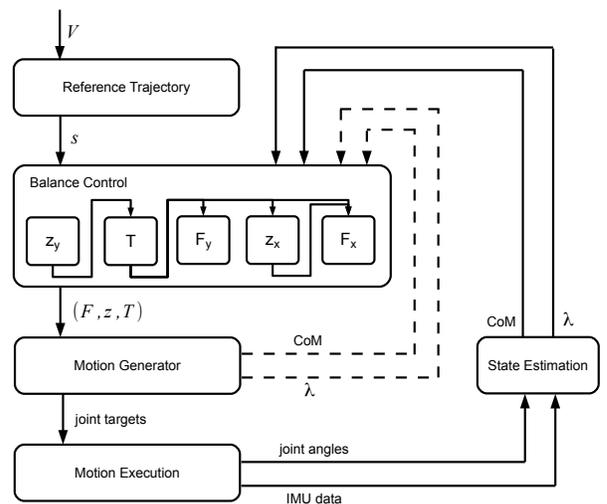


Fig. 3. Hierarchical structure of our gait control architecture.

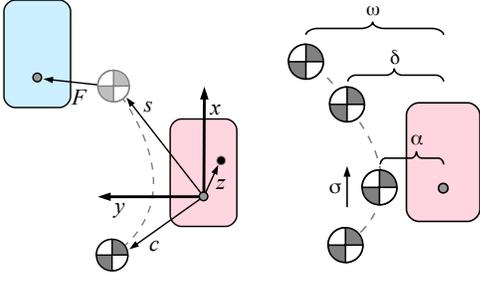


Fig. 4. **Left:** the notation used to describe the state of the system. The current CoM is denoted as c . The nominal CoM at the support exchange is labeled as s , z is the ZMP offset relative to the support foot, and F is the location of the next foot step relative to the future CoM at the end of the step. **Right:** the reference CoM trajectory is described by four configuration parameters that define the lateral distance at the step apex (α), the minimal and maximal support exchange locations (δ and ω), and the sagittal CoM velocity at the step apex for the maximum sagittal walking velocity (σ).

the model is currently standing on (λ) and the state of the center of mass are estimated.

Figure 3 indicates an inner loop that feeds back the CoM state and the support leg sign as assumed by the motion generation layer. The inner loop essentially drives the gait generation framework in open-loop mode, which in the best case should be able to generate a stable walk in the absence of disturbances. It is beneficial to achieve open-loop stability first, before switching to the outer loop and thereby allowing the state feedback to take control of the step parameters.

In the following, we will use the notation illustrated in Figure 4. The current CoM state is referred to as $c = (c_x, \dot{c}_x, c_y, \dot{c}_y)$ with sagittal and lateral coordinates and velocities with respect to the center of the current support foot. The balance controller steers c towards a nominal support exchange location $s = (s_x, \dot{s}_x, s_y, \dot{s}_y)$ using a zero moment point that is defined by an offset $z = (z_x, z_y)$ with respect to the center of the support foot. The next footstep location $F = (F_x, F_y)$ is expressed with respect to the CoM location. The time of the next support exchange is denoted as T . We use the symbol $\lambda \in \{-1, 1\}$ to encode the sign of the current support leg (left or right).

VI. REFERENCE TRAJECTORY GENERATION

The gait generation cycle begins with the computation of a reference CoM trajectory for the case of undisturbed, limit cycle motion of the linear inverted pendulum model. The entire reference trajectory is represented by the nominal support exchange state s . This support exchange state depends only on the velocity input V and configuration parameters, but not on the current state of the CoM.

Four configuration parameters α, δ, ω , and σ influence the shape of the nominal CoM trajectory and can be used to tune the open-loop characteristic of the walking motion. The meaning of these parameters is illustrated in Figure 4. The parameter α defines the lateral distance between the support frame and the CoM at the step apex, where the lateral CoM velocity \dot{c}_y is 0. δ defines the lateral support exchange location when the lateral walking velocity equals 0, while ω

defines the support exchange location for the maximal lateral walking velocity V_y^{max} . Finally, σ defines the sagittal CoM velocity \dot{c}_x at the step apex $c_x = 0$ for the maximum sagittal walking velocity V_x^{max} .

Given these configuration parameters and the walking velocity input V , the nominal support exchange state $s = (s_x, \dot{s}_x, s_y, \dot{s}_y)$ is computed with the following formulas:

$$s_y = \begin{cases} \lambda \xi, & \text{if } \lambda = \text{sgn}(V_y) \\ \lambda \delta, & \text{else} \end{cases}, \quad (10)$$

$$\dot{s}_y = \lambda C \sqrt{s_y^2 - \alpha^2}, \quad (11)$$

$$s_x = \frac{\sigma V_x}{C V_x^{max}} \sinh(C\tau), \quad (12)$$

$$\dot{s}_x = \frac{\sigma V_x}{V_x^{max}} \cosh(C\tau), \quad (13)$$

$$\xi = \delta + \frac{|V_y|}{V_y^{max}} (\omega - \delta), \quad (14)$$

$$\tau = \frac{1}{C} \ln \left(\frac{\xi}{\alpha} + \sqrt{\frac{\xi^2}{\alpha^2} - 1} \right), \quad (15)$$

where $\lambda \in \{-1, 1\}$ denotes the sign of the currently assumed support leg. The nominal support exchange state defines a target that the balance controller aims for and tries to reach at the end of the step. In (14) and (15) we computed meaningful quantities. The lateral support exchange location ξ is used when a step is taken in the direction of the lateral velocity V_y , and τ is the half step time that the CoM travels from the apex point to the support exchange location. The half step time will be useful for calculating the sagittal step size.

VII. BALANCE CONTROL

The underlying assumption we make to model the physical behavior of the robot is that its center of mass dynamics can be mathematically expressed with the equations of the linear inverted pendulum model [22]. Then, knowing the current state of the point mass, we can compute predictions of future states as well as suitable pivot points (step locations) to influence future trajectories in a way that they would follow reference trajectories that are known to result in a stable walk. Further assumptions we make are that the robot is always standing on exactly one leg, i.e. a double support phase is not explicitly modeled. At the time of the support exchange, the pendulum pivot point is set instantly to a new location, but the velocity of the CoM does not change and no energy loss occurs. The reason behind not including a double support phase in our model is for the sake of simplicity. Flat sole walkers do not roll their feet and they do not perform push off actuation. Hence, the double support is short and it has only negligible effect on CoM actuation. While making the model imperfect, these assumptions give us a simple and analytically feasible tool to design and control bipedal walking behavior.

The task of the Balance Control layer is to compute ZMP, timing and foot-placement parameters based on the currently measured CoM state and a target nominal state that the CoM

should reach in the moment of the support exchange. The step timing and the ZMP parameters are responsible for steering the CoM towards the target state during the current step, while the footstep location parameters are computed such that the CoM will return to the nominal trajectory during the next step without further control effort. It is not generally feasible to reach the target state by the end of the support phase. First of all, it is not always possible to connect two arbitrary states with a single zero moment point that stays constant during the entire support phase, and our controller does not attempt to solve the general case. On the other hand, even when a single ZMP location does exist, it may not be reachable if it lies outside of the support polygon. Therefore, the combination of ZMP and foot placement strategies is important to respond to strong perturbations.

A. Lateral ZMP Offset

Given the current CoM state $c = (c_x, \dot{c}_x, c_y, \dot{c}_y)$ and the nominal support exchange location $s = (s_x, \dot{s}_x, s_y, \dot{s}_y)$, the first parameter to compute is the lateral ZMP offset z_y . As long as the CoM does not cross the pivot point in the lateral direction, we can be certain that it will return and it will eventually reach the support exchange location s_y . But even after the slightest disturbance, it will not do so with the nominal support exchange velocity \dot{s}_y . Based on the constant orbital energy formula (8), we determine z_y such that the resulting ZMP would return the CoM to the nominal support exchange location with the nominal velocity

$$z_y = \frac{C^2 (s_y^2 - c_y^2) + \dot{c}_y^2 - \dot{s}_y^2}{2C^2 (s_y - c_y)}, \quad (16)$$

but bound it to be inside the support polygon and thus accept an error in the support exchange velocity. This error is corrected by choosing an appropriate lateral step size. Near the support exchange location where $c_y \approx s_y$, the lateral ZMP cannot be estimated due to a singularity in (16). Hence, we inhibit the ZMP adaptation shortly before and after the step. Having chosen a lateral ZMP offset, we can set the current CoM state to $c' = (c_x, \dot{c}_x, c_y - z_y, \dot{c}_y)$ so that subsequent computations will take the lateral ZMP offset into account seamlessly.

B. Step Time

We want the support exchange to occur when the CoM reaches the nominal support exchange location. Using the CoM state c' that already contains the lateral ZMP offset, the step time T is calculated using (4):

$$T = \frac{1}{C} \ln \left(\frac{s_y - z_y}{c_1} \pm \sqrt{\frac{(s_y - z_y)^2}{c_1^2} - \frac{c_2}{c_1}} \right), \quad (17)$$

$$c_1 = \left(c'_y + \frac{\dot{c}'_y}{C} \right), \quad (18)$$

$$c_2 = \left(c'_y - \frac{\dot{c}'_y}{C} \right). \quad (19)$$

All of the following step parameters depend on the step time T .

C. Lateral Step Size

The lateral step size F_y supports the return of the CoM to the reference trajectory during the next step. First, we use the CoM state c' , which now includes the lateral ZMP offset, and the estimated step time T to predict the actual future CoM velocity $\dot{y}(T)$ at the support exchange. Then we calculate the lateral footstep location such that the CoM will pass the apex of the next step at distance α with a velocity of zero. Note that the step size F_y is expressed with respect to the future CoM state $y(T)$:

$$\dot{y}(T) = c'_y C \sinh(CT) + \dot{c}'_y \cosh(CT), \quad (20)$$

$$F_y = \lambda \sqrt{\frac{\dot{y}(T)^2}{C^2} + \alpha^2}. \quad (21)$$

Essentially, we determined the limit cycle step size that would result in the same end-of-step velocity $\dot{y}(T)$ as predicted from the current CoM measurement.

D. Sagittal ZMP Offset

For the computation of the sagittal ZMP offset, we use the capture point based formula proposed by Engelsberger et al. in [13]. The main advantage of this approach is that it does not require a measurement of the actual zero moment point, which is relatively difficult to obtain from the kinematic model. It is sufficient to use the capture point as input, which requires only a first order derivation of the observed CoM trajectory:

$$z_x = \frac{s_x + \frac{\dot{s}_x}{C} - e^{CT} (c'_x + \frac{\dot{c}'_x}{C})}{1 - e^{CT}}. \quad (22)$$

This approach also does not guarantee that the CoM will arrive at the desired location at time T with the correct velocity, but it ensures that the capture point of the future CoM state equals the capture point of the desired state, which is a good approximation for our purposes. After bounding the sagittal ZMP offset z_x to remain inside the support foot, we integrate it into our CoM state estimation $c'' = (c'_x - z_x, \dot{c}'_x, c'_y, \dot{c}'_y)$, so that it will be taken into account seamlessly.

E. Sagittal Step Size

To obtain an adequate sagittal step size F_x , again we map the measured CoM state to a limit cycle that would result in the same end-of-step CoM velocity. Using the estimated step time T and the currently measured CoM state c'' , which already contains the effect of a disturbance as well as the ZMP offset, we predict the sagittal CoM velocity $\dot{c}_x(T)$ at the time of the support exchange. Then, we compute the nominal step size that would have resulted in the same velocity at the end of the step:

$$\dot{c}_x(T) = c''_x C \sinh(CT) + \dot{c}''_x \cosh(CT), \quad (23)$$

$$F_x = \frac{\dot{c}_x(T)}{C} \tanh(C\tau). \quad (24)$$

Please note that in (24) we used the half step time τ that was introduced in (15).

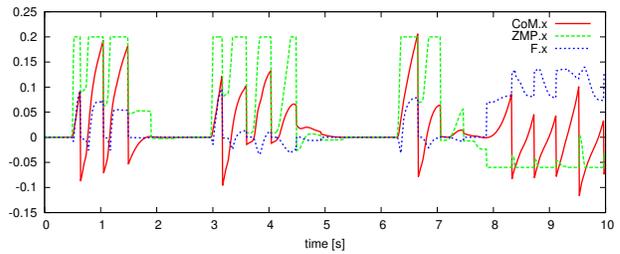
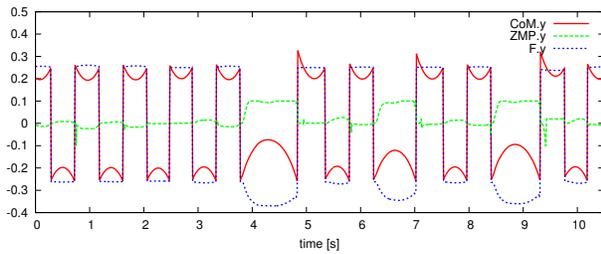


Fig. 5. CoM, ZMP and footstep coordinate data recorded from the simulated robot during a pushing experiment. The robot was pushed three times from the side (left) and returned each time to its nominal gait pattern after one step. Then the robot was pushed from the back (right) and recovered its balance after taking several capture steps.

VIII. MOTION PATTERN GENERATION

The core of our walking motion generator is a LIP model that produces continuous CoM trajectories. In each iteration of the control framework, we perform the following steps to generate an increment of the CoM position. First, we shift the current pivot point by the ZMP parameters from the Balance Control layer and compute a motion increment Δt of the pendulum. Then we integrate the sensor feedback from the state estimation into the motion pendulum state

$$c_{k+1} = (1 - \xi)c_k(\Delta t) + \xi\hat{c}, \quad (25)$$

where $c_k(\Delta t)$ is the computed CoM location Δt time later, \hat{c} is the CoM state as it was measured from the kinematic model, and $0 \leq \xi \leq 1$ is a smoothing parameter. The CoM position sequence is used as the input into inverse kinematics calculations that determine the joint angles for the support leg to follow the desired motion. At this point, we assume that the resulting ZMP will be near the desired ZMP and we do not attempt more precise ZMP regulation.

For the swing leg, we generate a spline based motion trajectory in Cartesian space that starts with the last commanded state of the swing foot and ends with the new footstep location and zero velocity at time T . This way, sudden changes in the footstep location can be smoothly integrated. Again, we use inverse kinematics to generate the necessary joint angles to realize the desired swing motion. If the step time T is near zero, or premature floor contact has been detected, we switch the roles of the legs between support leg and swing leg. The joint angles are sent to the robot after each framework iteration.

Computing only one increment of the pendulum motion with a known ZMP is evidently much faster than optimizing the CoM trajectory to track a future ZMP reference, for example with the model predictive control (MPC) [6] algorithm. Unlike MPC, we do not attempt to minimize the jerk. Unbounded jerk, however, can only occur in the moment of the support exchange. While the CoM positions and velocities are still continuous at all times, we believe that this is not an issue on real hardware. Not using a future ZMP reference has the additional benefit that the target footstep location and the ZMP offset can change instantaneously at any time during the step, for example as a result of a sudden disturbance. The CoM and swing foot trajectory generators will continue to produce smooth position commands without

any additional computation.

While our implementation of the motion pattern generator works together with simulated high gain position controlled motors, in principle, any motion generation algorithm can be used in conjunction with our balance control layer, as long as it allows parameterized input of footstep coordinates, timing, and a ZMP offset. For example, torque control based algorithms for elastic actuators, such as Virtual Model Control [23], can also be potentially combined with our method.

IX. EXPERIMENTAL RESULTS

To demonstrate the efficiency of our balance controller, we performed push experiments on a simulated humanoid robot with a total body weight of 13.5 kg and a roughly human-like mass distribution. We used a simulation software based on the Bullet physics engine [24]. The collected data from the simulated robot is visualized in Figure 5. In our first experiment, we push the robot with an impulse of 3 Ns from the side while it was walking on the spot. The first push occurs approximately at the time mark of 3.8 s, where a strong disturbance in the CoM trajectory is clearly visible. The lateral ZMP offset increases quickly to the limit of 10 cm. At the same time, the lateral step size increases to counteract the disturbance during the next step. The robot successfully returns to its nominal oscillation amplitude after one step, before two steps later the robot is pushed from the side again.

In the second experiment, we push the robot three times from the back with an impulse of 6 Ns while it is walking on the spot. The first push occurs at the time mark of 0.5 s. The CoM trajectory data indicates how the robot suddenly starts to move forward after the push. The sagittal ZMP is immediately shifted to the toe of the robot at 20 cm. The step size is also increased and the robot performs four capture steps with decreasing size to regain its balance and come to a halt again. The robot is pushed two more times and successfully regains balance after taking several steps. After the last push, the robot starts walking forward. The accompanying video shows how these experiments were performed in simulation.¹

It is difficult to assess the stability of our controller in a simple way. The magnitude of an impulse that can be absorbed varies strongly with the time of the gait phase, the location of the contact point on the body and the direction

¹ http://www.ais.uni-bonn.de/movies/Humanoids_2013_Omnidirectional_Capture_Steps.wmv

of the impact. For example, much stronger pushes can be absorbed in sagittal direction than in lateral direction.

We have measured a total execution time of 0.12 ms of the entire control loop with unoptimized code when executed on a 1.3 GHz core. This includes filtering of the CoM input, the step parameter calculations in the Balance Control layer and the joint trajectory generation in the Motion layer.

X. CONCLUSIONS

We have presented a bipedal locomotion framework that simplifies the implementation of a closed-loop walk by decomposing the task into a linear inverted pendulum based balance controller and a motion generator that interface using a set of step parameters, such as the location and timing of the next step and the desired ZMP. We demonstrated the ability of the framework to generate an omnidirectional walk and to cope with strong disturbances in simulation. The main conclusion to be drawn is that despite the radical simplification of the whole-body dynamics to an uncoupled two-dimensional point mass model, the feedback control loop is able to recover from disturbances that are strong enough to force the robot to take recovery steps.

A clear distinction between our method and most of other algorithms used to date is that we do not attempt to follow a future ZMP reference. Instead, we express motion trajectories as a limit cycle CoM reference, from which step parameters arise naturally and can be flexibly changed as a response to a disturbance. The control laws are analytically derived from the mathematical model and can be calculated in closed form.

In future work, we will exchange our motion generator with an implementation of a more natural gait with stretched knees and torso actuation. We will incorporate hip torque strategies, using the torso mass for more efficient disturbance rejection, and we will continue to investigate methods to cope with floor inclination and angular momentum about the edge of the support foot. Furthermore, we will extend our algorithm with learning capabilities that allow us to improve the efficiency of capture steps by learning the difference between the simplified physical model and the real hardware and automate the learning process by allowing the robot to autonomously explore using self induced disturbances.

REFERENCES

- [1] Sven Behnke. Online trajectory generation for omnidirectional biped walking. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1597–1603, 2006.
- [2] Hao Dong, Mingguo Zhao, and Naiyao Zhang. High-speed and energy-efficient biped locomotion based on virtual slope walking. *Autonomous Robots*, 30(2):199–216, 2011.
- [3] Kazuo Hirai, Masato Hirose, Yuji Haikawa, and Toru Takenaka. The development of honda humanoid robot. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 1998.
- [4] Shuuji Kajita, Mitsuharu Morisawa, Kanako Miura, Shinichiro Nakaoka, Kensuke Harada, Kenji Kaneko, Fumio Kanehiro, and Kazuhito Yokoi. Biped walking stabilization based on linear inverted pendulum tracking. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 4489–4496, 2010.
- [5] Ill-Woo Park, Jung-Yup Kim, Jungho Lee, and Jun-Ho Oh. Mechanical design of humanoid robot platform khr-3 (kaist humanoid robot 3: Hubo). In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, pages 321–326, 2005.
- [6] Pierre-Brice Wieber. Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, pages 137–142, 2006.
- [7] Marc Raibert, Kevin Blankespoor, Gabriel Nelson, Rob Playter, and the BigDog Team. BigDog, the rough-terrain quadruped robot. In *Proc. of the 17th World Congress*, 2008.
- [8] Ryosuke Tajima, Daisaku Honda, and Keisuke Suga. Fast running experiments involving a humanoid robot. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2009.
- [9] Baek-Kyu Cho, Sang-Sin Park, and Jun ho Oh. Stabilization of a hopping humanoid robot for a push. In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, pages 60–65, 2010.
- [10] Mitsuharu Morisawa, Fumio Kanehiro, Kenji Kaneko, Nicolas Mansard, Joan Sola, Eiichi Yoshida, Kazuhito Yokoi, and Jean-Paul Laumond. Combining suppression of the disturbance and reactive stepping for recovering balance. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3150–3156, 2010.
- [11] Benjamin Stephens. Humanoid push recovery. In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2007.
- [12] Junichi Urata, Koichi Nishiwaki, Yuto Nakanishi, Kei Okada, Satoshi Kagami, and Masayuki Inaba. Online decision of foot placement using singular lq preview regulation. In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, pages 13–18, 2011.
- [13] Johannes Engelsberger, Christian Ott, Máximo A. Roa, Alin Albu-Schäffer, and Gerhard Hirzinger. Bipedal walking control based on capture point dynamics. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 4420–4427, 2011.
- [14] Jerry E. Pratt, Twan Koolen, Tomas de Boer, John R. Reubla, Sebastian Cotton, John Carff, Matthew Johnson, and Peter D. Neuhaus. Capturability-based analysis and control of legged locomotion, part 2: Application to m2v2, a lower-body humanoid. *The International Journal of Robotic Research*, 31(10):1117–1133, 2012.
- [15] John Reubla, Fabian Canas, Jerry Pratt, and Ambarish Goswami. Learning capture points for humanoid push recovery. In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2007.
- [16] Marcell Missura and Sven Behnke. Lateral capture steps for bipedal walking. In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2011.
- [17] Juan José Alcaraz-Jiménez, Marcell Missura, Humberto Martínez Barberá, and Sven Behnke. Lateral disturbance rejection for the nao robot. In *RoboCup*, pages 1–12, 2012.
- [18] Arthur D. Kuo, J. Maxwell Donelan, and Andy Ruina. Energetic consequences of walking like an inverted pendulum: step-to-step transitions. *Exercise and Sport Sciences Reviews*, 33(2):88–97, 2005.
- [19] Marcell Missura and Sven Behnke. Dynaped demonstrates lateral capture steps. <http://www.ais.uni-bonn.de/movies/DynapedLateralCaptureSteps.wmv>.
- [20] Arthur D. Kuo. Stabilization of lateral motion in passive dynamic walking. *The International Journal of Robotics Research*, 18(9):917–930, 1999.
- [21] Catherine E. Bauby and Arthur D. Kuo. Active control of lateral balance in human walking. *Journal of Biomechanics*, 33(11):1433–1440, 2000.
- [22] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kazuhito Yokoi, and Hirohisa Hirukawa. The 3D linear inverted pendulum mode: a simple modeling for a bipedwalking pattern generation. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 239–246, 2001.
- [23] Jerry Pratt, Chee-Meng Chew, Ann Torres, Peter Dilworth, and Gill Pratt. Virtual model control: An intuitive approach for bipedal locomotion. *The International Journal of Robotics Research*, 20(2):129–143, 2001.
- [24] Adrian Boeing and Thomas Bräunl. Evaluation of real-time physics simulation systems. In *Proc. of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia, GRAPHITE '07*, pages 281–288, New York, NY, USA, 2007. ACM.