

# Lateral Capture Steps for Bipedal Walking

Marcell Missura and Sven Behnke

**Abstract**—Bipedal walkers are difficult to control, inherently unstable systems. Besides the complexity of the walking motion itself, the balance of the robot constantly has to be maintained with good foot placements and other disturbance-rejection strategies. In this work, we are presenting a new, closed-loop control approach that addresses both, the problem of complexity and the challenge of maintaining balance during walking. We decouple walking motion from balance and combine them in a hierarchical framework allowing a foot placement-based balance regulator to control the timing and footstep coordinates of central pattern-generated stepping motions. Furthermore, we decompose the balance controller into three simple, independent modules that compute suitable estimates of timing and sagittal and lateral coordinates for the next footstep to maintain a nominal center of mass trajectory.

We implemented the timing and the lateral step size components using the equations of a parameterized version of the linear inverted pendulum model that we fit to data collected from a walking robot. The parameter optimization has a significant impact on the accuracy of our predictions. We demonstrate the efficiency of our approach by performing experiments on a real biped. Results show that the robot is able to reliably recover from any lateral push in only a few steps as long as it does not tip over the current support leg.

## I. INTRODUCTION

For decades, it has been a dream of scientists and engineers to replicate the efficiency, stability, and grace of the natural human gait. But the analytic calculation of dynamic and balanced full-body motions is a difficult task. The applicability 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.

We propose a lightweight bipedal walking framework and use it to implement a closed-loop control algorithm that maintains the lateral balance of a humanoid robot (Fig. 1). By adjusting the timing and the location of the next step, our approach is able to recover from strong lateral disturbances and to continue walking with a nominal open-loop frequency after only a few capture steps. A precise dynamic model of the robot is not required, nor the calculation of the Zero Moment Point. Furthermore, our algorithm requires very little computational power and specifically addresses the issues of sensor noise and latency.

Our strategy to tackle complexity is to separate motion from balance and to decompose the balance controller itself to calculate the timing of the steps and the lateral and sagittal step coordinates in independent modules. Each of these components is much simpler to solve when isolated from

All authors are with the Autonomous Intelligent Systems Group, University of Bonn, Germany. Email: missura@ais.uni-bonn.de. This work has been supported partially by grant BE 2556/2-3 of German Research Foundation (DFG).

the others. For the generation of stepping motions, we use an open-loop central pattern-generated omnidirectional gait. The motion patterns are parameterized in step frequency and leg-swing amplitudes such that they can easily be modified by a higher layer, which is in this case our footstep control algorithm.

To derive the components of the footstep controller, we reduce our multi body system to a single point mass whose physical dynamics is described by a parameterized version of the well-known linear inverted pendulum model (LIPM). We fit the model parameters to maximize the similarity with center-of-mass trajectory data collected from a real robot and achieve a significant increase in accuracy, compared to the standard LIPM. Using the fitted model, we can analytically predict suitable capture steps, which are able to maintain lateral balance out of the box, without further parameter tuning. To deal with sensor noise, we follow an open-loop gait trajectory as much as possible, unless a significant deviation is detected. In this case, the control loop adapts the next footstep only within a relatively noise-free time window during mid-swing, and continues with open-loop execution until the next mid-swing phase, where we can decide if further corrections are necessary.

The only requirements for our algorithm are joint angle sensors, an inertial measurement unit and a kinematic model of the robot. Even low-cost robots are typically equipped with these kind of sensory systems. A kinematic model is also often available or can be easily obtained. Foot pressure sensors can be useful to detect footsteps, but one can also implement a footstep detector based on the kinematic model. The robot we used is not equipped with foot sensors.

The remainder of this paper is organized as follows. After



Fig. 1. Our robot Dynaped rejects lateral disturbances with capture steps.

reviewing related work, we begin with an analysis of the pendulum-like biped walking mechanics in Section III to motivate our decomposition approach. Then we introduce our concept of hierarchical decomposition of bipedal gait control in Section IV. In Section V, we briefly review the linear inverted pendulum model, which is the core element of our balance control layer. After deriving general lateral control laws from observing the model behavior in Section VI, we focus on fitting the pendulum model to our robot in Section VII. In Section VIII, we discuss our approach to dealing with sensor noise and Section IX summarizes experimental results.

## II. RELATED WORK

Zero Moment Point-based trajectory tracking walkers with position-controlled joints are arguably the most successful so far. ASIMO [7], HRP-2 [8], and HUBO [14] are among the most prominent examples. 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 future steps with a fixed frequency prevents a flexible response to strong perturbations that require a change of footstep locations and timing in order to maintain balance.

Also very impressive are the methods used by the leading teams of the annual RoboCup soccer competitions (e.g. [5], [6], [2]), where hand-tuned rhythmical patterns produce a self-stable, open-loop gait relying on no more than a narrow basin of attraction around a fix point for stability. This is enough to play dynamic and exciting soccer games on a flat and hard surface, but collisions and other disturbances lead to falls in almost all cases. The central pattern-generated omnidirectional gait engine that we used to implement parameterized stepping motions is a well-proven RoboCup method [2].

To recover from strong pushes and collisions, reactive stepping is necessary, as demonstrated by the amazing performance of the quadruped BigDog [15]. Among bipedal walkers, reactive stepping is a new discipline. Toyota’s new running robot [19] and HUBO [3] demonstrated the ability to cope with a frontal push against the chest during hopping on the spot. MABEL can also be disturbed during walking and the steps will be adjusted to regain stability [17]. However, this robot is mounted to a beam and only planar walking has been investigated. A popular approach is to combine momentum suppression and reactive stepping [3] [13] [18] such that a reactive step only needs to be taken if the disturbance cannot be compensated otherwise. From these proposals only the approach of Morisawa et al. [13] is able to react to a push from any direction at the cost of complex calculations – despite the simplified dynamic model. Their method also requires an estimation of external forces, it does not attempt to adjust step timing and the efficiency is not yet reliable enough to operate the robot in an unconstrained environment.

Lateral walking itself received very little attention, even though it appears to be more challenging, compared to walking in the sagittal plane. Kuo investigated passive lateral stability

in [10] and came to the conclusion that active lateral stabilization is necessary. Bauby and Kuo [1] found that the lateral step size variability of humans drastically increases when the eyes are closed, which is strong evidence that lateral stability is more sensitive to sensor feedback, or the lack thereof.

The closest related work was presented by Graf and Röfer [6], who proposed a LIPM-based closed-loop gait similar to our concept. As one of very few examples aside from our approach, the timing of the steps is also taken into account. However, while in our approach the timing is exclusively determined by lateral oscillation, in [6] the timing of the steps is synchronized with sagittal stepping instead. Lateral step sizes are not considered. While [6] makes heavy use of sensor feedback, our strategy is to remain open-loop as much as possible. Also, our predictions are calculated analytically and thus are potentially faster than the numerical method used in [6].

Our capture steps are similar to the capture point [16] introduced by Pratt et al. as the point where a robot has to step in order to stop. We extend the definition of the capture point to a more flexible capture step coordinate that allows to continue walking with the desired velocity even after a disturbance. The linear inverted pendulum model was first proposed in [9]. Our balance control layer is based entirely on this model. However, we extend the model to a parameterized version to better approximate the dynamics of our robot.

Aside from the concentration on lateral stability, a clear distinction between our method and the cited publications is that our algorithm makes do without a precise physical model of the robot, i.e. masses, joint torques and external forces do not need to be known. We use only a small set of linear inverted pendulum equations to predict the trajectory of the center of mass. The location of the Zero Moment Point is not needed. This makes our approach simple, robust and easy to work with while still providing an effective balancing strategy.

## III. MOTIVATION

The pendulum-like mechanics of human walking has been long known to be an energy-efficient physical principle [11]. Figure 2 shows stick diagrams of the idealized sagittal and lateral pendulum motions projected on the sagittal plane (left) and the frontal plane (right), respectively. Interestingly, the nature of the sagittal and the less frequently analyzed lateral motion exhibit an opposing conceptual difference. In the sagittal plane, the center of mass “vaults” over the support

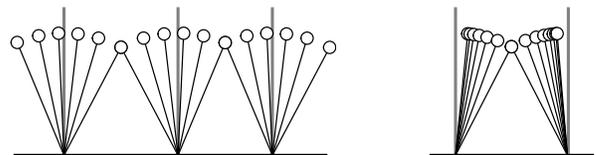


Fig. 2. Stick diagrams of the idealized pendulum-like sagittal motion (left) and lateral motion (right) of a compass gait. In sagittal direction, the center of mass passes over the pendulum pivot point in every gait cycle, while in lateral direction it is crucial that the pendulum never crosses the pivot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

leg 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 sideways over the support leg, as the recovery from such an unstable state requires a combination of crossing the legs, a sharp turn around the stance foot, and in extreme cases even a jump to free the support leg and place it at a more convenient location. Humanoid robots have difficulties with performing such maneuvers. Considering that the lateral distance between the center of mass and the pivot point at the apex of the step provides only a narrow margin for errors, it is not surprising that substantially more effort needs to be invested in lateral control [1].

Other than a shared time of support exchange, the sagittal motion and the lateral motion do not appear to have an evident influence on each other. In Figure 3, we show center of mass states in the phase space of the lateral motion recorded from our Dynaped. Three data sets are plotted: one for undisturbed walking on the spot, one for walking forward at maximum velocity, and a set of unstable lateral swaying. One can observe that forward walking has only a small influence on the lateral trajectories, especially in comparison with lateral disturbances. This observation supports our assumption that sagittal and lateral controllers can be modeled independently.

Furthermore, we assume the lateral center of mass oscillation to be the primary determinant of the step timing. While in sagittal direction, the biped can flexibly respond to variations in timing with a change of the stride length, in the more sensitive lateral direction support exchange should always occur when the center of mass is in the middle of the stride to sustain a stable lateral oscillation at a steady frequency.

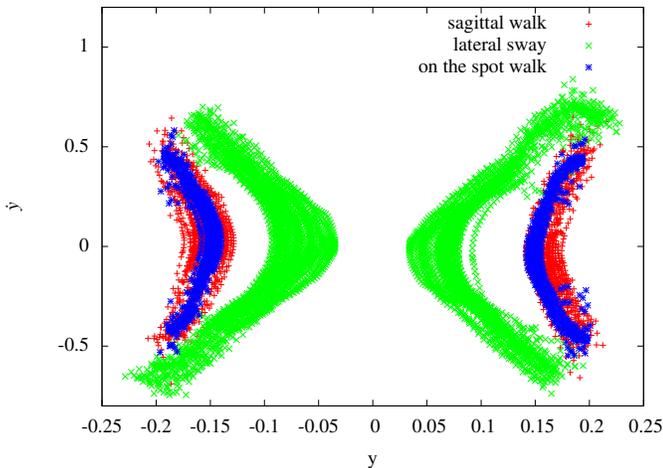


Fig. 3. Center of mass trajectories in the phase space of the lateral pendulum model. The red data set shows CoM states collected during forward walking. The blue data set was recorded during undisturbed walking on the spot. The green data set shows trajectories of unstable lateral swaying. The lateral state trajectories during walking on the spot and during forward walking are very similar. The data is noisy at the end points of the “boomerangs”, where the support exchange occurs.

## IV. HIERARCHICAL DECOMPOSITION

### A. Control Interface

Our bipedal locomotion framework is composed of three hierarchical layers, as illustrated in Figure 4. The top layer is an abstract omnidirectional control interface that can be used by high-level behaviors to navigate the robot to desired goals. At this level of abstraction, the robot is assumed to be a holonomic point mass, controlled with a three-dimensional gait velocity vector  $V \in \mathbb{R}^3$  in sagittal, lateral, and rotational directions. The velocity vector is an intuitive control input in SI units that eliminates the need to be concerned with single steps or even joint angles. The control interface converts the velocity vector  $V$  to an appropriate combination of step frequency  $\phi^*$  and step size  $S^* = (S_x^*, S_y^*, S_z^*) \in \mathbb{R}^2 \times [-\pi, \pi]$ , where  $S_x^*$  and  $S_y^*$  are the desired sagittal and lateral Cartesian coordinates of the footstep, expressed in the reference frame of the current support foot, and  $S_z^*$  is a rotation of the feet. We will refer to  $\phi^*$  and  $S^*$  as desired step parameters.

### B. Foot Placement Control

The middle layer is the foot placement control layer. Its purpose is to maintain the balance of the biped while respecting the reference velocity as much as possible without becoming unstable. In addition to the desired step parameters  $\phi^*$  and  $S^*$ , this layer receives input from a module that we call kinematic-balance-model (KB-model). The KB-model applies the joint angles received from the robot to the kinematic model using forward kinematics and rotates the entire model around the center of the support foot such that the trunk has the angle reported by the inertial measurement unit. From the rotated model, we extract the sagittal and lateral distances and velocities of the center of mass (CoM) with respect to the current support foot and obtain a four-dimensional CoM state  $(x, \dot{x}, y, \dot{y})$ , where  $x$  and  $\dot{x}$  denote the sagittal center of mass location and velocity, and  $y$  and  $\dot{y}$  denote the lateral center

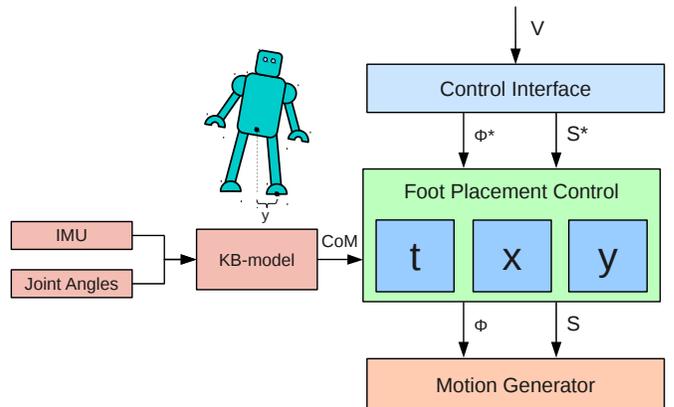


Fig. 4. Hierarchical structure of our gait control architecture. The Foot Placement Control layer receives the center of mass (CoM) state with respect to the support foot from the kinematic balance model. The step parameters from the Control Interface can be modified by the modules of the Foot Placement Control layer in order to maintain balance. The lowest layer is a parameterized central pattern-based motion generator for stepping motions.

of mass location and velocity. The CoM state is a simple, low-dimensional representation of balance, sufficient for the foot placement layer to detect instabilities and to estimate footstep timings and coordinates. As a consequence of our assumptions in Section III, the internal structure of the foot placement control layer is horizontally decomposed into three independent control units that determine the time  $t$ , the sagittal step size  $S_x$ , and the lateral step size  $S_y$  of the next balanced footstep in parallel. The implementation of these components will be discussed in detail in Section VI.

### C. Motion Generator

The bottom layer generates stepping motions using the central pattern generator-based method described in [2]. The step size  $S$  as input from the foot placement control layer defines the leg swing amplitudes and the gait frequency  $\phi$  determines the timing. This is all the information needed to generate the next step. Thus, the motion generation layer does not need to be concerned with balance. 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 and timing.

## V. THE LINEAR INVERTED PENDULUM MODEL

The linear inverted pendulum model is an approximation of the principle dynamics of human walking [9]. In its simplest form, it describes a motion in one dimension governed by the equation

$$\ddot{x} = Cx, \quad (1)$$

where  $C$  is a gravitational constant. Given an initial state  $(x_0, \dot{x}_0)$ , the set of equations

$$x(t) = c_1 e^{\sqrt{C}t} + c_2 e^{-\sqrt{C}t} \quad (2)$$

$$\dot{x}(t) = c_1 \sqrt{C} e^{\sqrt{C}t} - c_2 \sqrt{C} e^{-\sqrt{C}t} \quad (3)$$

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

$$t(\dot{x}) = \frac{1}{\sqrt{C}} \ln \left( \frac{\dot{x}}{2c_1 \sqrt{C}} \pm \sqrt{\frac{\dot{x}^2}{4c_1^2 C} + \frac{c_2}{c_1}} \right) \quad (5)$$

$$c_1 = \frac{1}{2} \left( x_0 + \frac{\dot{x}_0}{\sqrt{C}} \right) \quad (6)$$

$$c_2 = \frac{1}{2} \left( x_0 - \frac{\dot{x}_0}{\sqrt{C}} \right) \quad (7)$$

allows analytic calculation of the state  $(x, \dot{x})$  at a given time  $t$  ((2), (3)), or of the time  $t$ , when the center of mass will reach a given position  $x$  (4) or a given velocity  $\dot{x}$  (5). Furthermore, the law of conservation of energy is valid. Unless the pendulum is disturbed by external forces, the orbital energy

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

is constant for an entire trajectory.

To model sagittal and lateral motion, we use a set of uncoupled equations for each of the two dimensions, respectively:

$$\ddot{x} = Cx, \quad (9)$$

$$\ddot{y} = Cy. \quad (10)$$

In this model, the motion in sagittal ( $x$ ) direction does not depend on the motion in lateral ( $y$ ) direction and vice versa. They can be treated as two orthogonally superimposed, one-dimensional pendulum motions.

## VI. DERIVATION OF CONTROL LAWS

The analysis of the model behavior leads directly to control laws that we can utilize to implement the balance control components. Figure 5 gives a “top down” view of our two-dimensional model to illustrate its concepts.

We define the nominal center of mass trajectory as the trajectory generated by the open-loop walk in the absence of disturbances. For now, we assume a constant sagittal velocity and concentrate only on the lateral case. The lateral component of our bipedal walking model is characterized by a perpetual oscillation of the center of mass in between two feet alternating the role of support. Figure 5(a) illustrates the trajectory of a nominal step. The step starts when both legs are in a symmetric configuration in the moment of the support exchange. We denote the distance between the CoM and the support foot in this moment with  $\delta$ . As the step continues, the center of mass approaches the support foot and reaches the closest proximity  $\alpha$  at the apex of the step. Here, the sign of the lateral velocity changes and the center of mass returns to the gait center  $\delta$ . The support exchange can occur anywhere between  $\delta$  and the maximal CoM location  $\omega$ , depending on the commanded lateral velocity from the higher layer. The next footstep coordinate is chosen such that the CoM velocity vanishes again at a distance  $\alpha$  to the next support foot. We model the support exchange as instantaneous and collisionless, so that the center of mass velocity remains continuous. Other than these “steps”, the system is driven only by its own passive dynamics.

An important observation to be made is that since the orbital energy (8) stays constant on every undisturbed CoM trajectory, and all our nominal steps have the same representative lateral state  $(\alpha, 0)$  even if the robot is walking in lateral direction, the lateral energy level remains constant at all times at a nominal value of  $E_n = -\frac{C}{2}\alpha^2$ . A deviation from this energy level in any direction is undesirable and needs to be corrected.

If a disturbance modifies the CoM trajectory, the step apex will not be reached at distance  $\alpha$  and we can categorize the trajectory as unstable. There are two possible cases that can result from a disturbance. In the better case, the center of mass will still turn back and eventually reach the support exchange location currently desired by the higher layer. This case is shown in part (b) of Figure 5. In the worse case, the CoM will cross the support foot and the robot will most likely fall. If a non-returning trajectory is detected early on in a gait cycle, the remaining time is better invested in preparing a damage-rejecting fall sequence [4]. Any returning CoM trajectory,

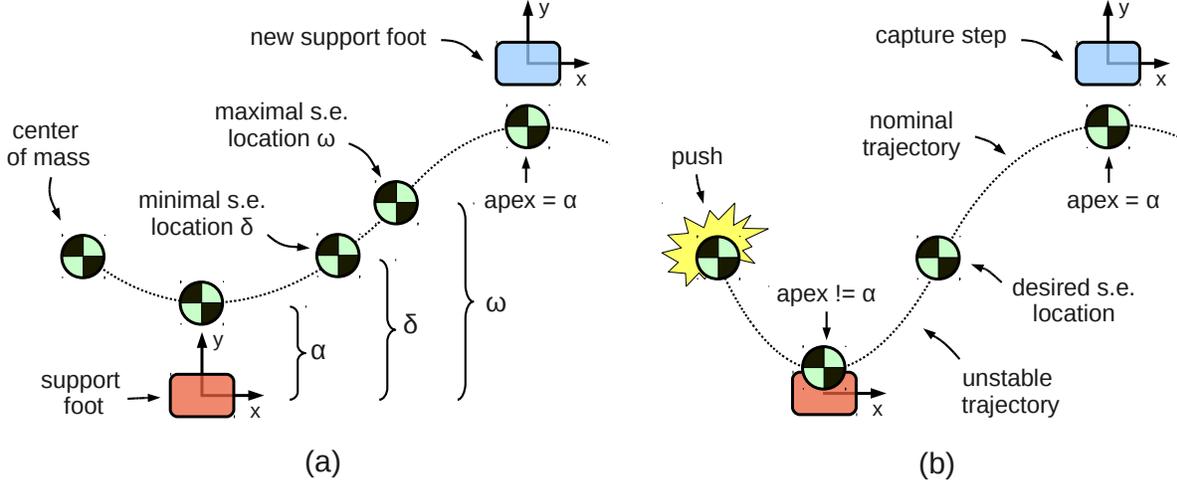


Fig. 5. Analysis of the lateral LIPM behavior. (a) Three parameters  $\alpha$ ,  $\delta$ , and  $\omega$  describe the closest foot proximity, the minimal support exchange (s.e.) location, and the maximal support exchange location, respectively. (b) A push can defer the center of mass trajectory to an unstable one, but as long as the robot does not tip over the support leg, a capture step can always be calculated such that the center of mass returns to a nominal trajectory.

however, can easily be captured, as it requires only a simple adjustment of step timing and step size.

In summary, the following lateral control law can be distilled. The time of the support exchange is the time when the center of mass will reach the currently desired support exchange point  $y_s$  between  $\delta$  and  $\omega$ . This accounts for maintaining the currently desired walking velocity even if the robot was pushed. At any point of the gait cycle, the estimated step time can be computed by setting  $(y_0, \dot{y}_0)$  to the current CoM state and using the desired support exchange location  $y_s$  as input for Equation (4).

To calculate the lateral coordinate of the capture step, the CoM velocity  $\dot{y}_s$  at the support exchange location is needed. It follows directly from (8):

$$\dot{y}_s = \sqrt{\dot{y}_0^2 + C(y_s^2 - y_0^2)}. \quad (11)$$

Now, using  $(\alpha, 0)$  as a representative state at the apex of the next step, the lateral distance  $y_c$  between the center of mass at the support exchange location and the new support foot can be calculated using again the energy equation (8):

$$y_c = \sqrt{\alpha^2 + \frac{\dot{y}_s^2}{C}}. \quad (12)$$

The lateral footstep coordinate  $S_y$  in the reference frame of the current support foot is then given by

$$S_y = y_s + y_c. \quad (13)$$

Please note that the energy formula does not depend on time. Hence, the lateral footstep coordinate can be calculated without knowing the time of the support exchange.

## VII. MODEL FITTING

Let us now concentrate on finding suitable pendulum parameters to match the physical behavior of a robot. As initially stated, our starting point is a readily available open-loop gait

that can walk stable on flat terrain without disturbances. This gives us easy access to data by calculating the KB-model during walking and recording the low-dimensional CoM state representation, introduced in Section IV.

Instead of setting  $C = \frac{G}{h}$ , as originally suggested in [9], we allow not only an arbitrary gravitational constant  $C$ , but also a lateral offset  $\Delta y$  from the support foot.

$$\ddot{y} = C(y + \Delta y), \quad (14)$$

We performed a grid search on this two-dimensional parameter space and found that choosing  $C = 10.33/s^2$  and  $\Delta y = 0.16m$  significantly improves the accuracy of the predictions that can be made with the pendulum model.

In Fig. 6, we show the mean squared error of the step time estimations using the default LIPM and our LIPM with the fitted parameters. The large difference between the suggested and our value for  $C$ , as well as the relatively high lateral offset  $\Delta y$  that places the pendulum pivot point outside of the support polygon, are not surprising. A humanoid robot is a complex kinematic chain. Its system dynamics must surely deviate from the linearized dynamics of a point mass. Also, our robot has relatively large and flexible feet, which further distort the physical behavior, compared to the point feet of a pendulum. Finally, while the inverted pendulum model assumes pure passive dynamics, the robot is actuated at all times and actively influences its motion during walking.

The KB-model reports CoM states relative to the support foot, without the offset  $\Delta y$ . Before using the pendulum equations for footstep estimation, we apply a mapping by adding the offset to the lateral location  $y$  with the adequate sign, depending on which leg is the support leg. The offset is later subtracted again from the result to obtain footstep coordinates in the reference frame of the actual support foot.

In addition to the pendulum parameters  $C$  and  $\Delta y$ , the CoM

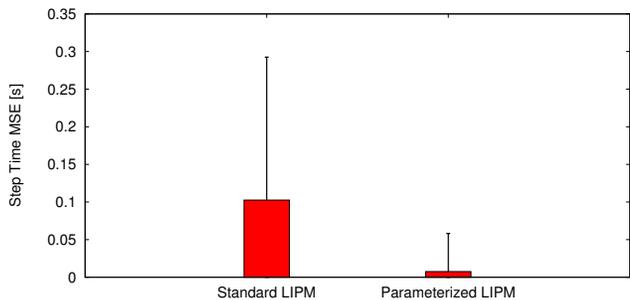


Fig. 6. Mean squared error and standard deviation of estimated step times by the default LIPM (left) and our parameterized pendulum (right). Fitting the model parameters to the data significantly improves the prediction accuracy. The mean step time of our robot is 0.43 seconds.

location parameters  $\alpha$ ,  $\delta$ , and  $\omega$  can also be extracted from the recorded data, so there is no need for manual tuning.

### VIII. NOISE SUPPRESSION

An essential part of the capture step algorithm is a filter that we apply to suppress sensor noise. The sensor noise increases right after support exchange and the CoM state cannot be estimated with sufficient precision in this phase of the step. The reason for this is a sensitive reaction of the MEMS acceleration and angular rate sensors to the vibrations caused by the collision of the feet with the ground.

Furthermore, the calculation of the speed of the stepping motions to match the anticipated step times is inherently unstable. We calculate a motion speed factor  $\sigma$  as

$$\sigma = \frac{t_m}{t_s} \quad (15)$$

with the remaining motion time  $t_m$  and the remaining time until the support exchange  $t_s$ , and use it to adapt the motion execution speed to the remaining time until the swing foot is expected to touch down.  $t_m$  and  $t_s$  both converge to zero towards the end of the step cycle. The remaining time until the step is, however, only an estimate that always contains an error  $\epsilon$ , which does not converge to zero. Therefore, the impact of the error increases towards the end of a step. The best time to estimate the motion speed is early in a swing phase. To deal with both sources of noise at once, we apply a simple Gaussian function  $f(\phi)$  with the mean  $\mu = 0.5$  set into the center of the swing phase  $0 \leq \phi < 1$  and a standard deviation of  $\sigma = 0.2$ , as depicted in Figure 7. These are the only parameters that we set manually, but they could also be automatically determined by analyzing the noise distribution with respect to the step phase. The Gaussian noise suppression function defines how much we adapt to modified footstep coordinates and timings calculated by the balance regulators. At the beginning of the step, where the sensor noise is high, and towards the end of the step, where the motion speed estimation is unstable, the adaptation rate is low. In the middle of the step, the adaptation rate is allowed to be high.

Aside from the Gaussian filter, we maintain an expected energy level  $E_e$  during the step. By default, this is set to

the nominal orbital energy  $E_n$ . Adaptation to modified step parameters only takes place if the deviation from the expected energy level is significant, for example as a result of a push. Then the expected energy  $E_e$  itself is also adapted to the new energy level  $E_m$  measured by the sensors. As long as the center of mass keeps traveling on a trajectory with a constant energy level, there is no need to continue adaptation after the step parameters have been adjusted.

The complete noise suppression algorithm that calculates the adaptation rate  $\rho$  is implemented as

$$\begin{aligned} \Delta E &= \|E_e - E_m\|, \\ \rho &= \Delta E \cdot f(\phi), \\ E_e &= \rho E_m + (1 - \rho) E_e. \end{aligned}$$

After a footstep has been detected, we reset the gait frequency, the step size, and the expected energy to their nominal values. The capture steps are calculated in a way that one step should be enough to restore stability, so the best strategy is to anticipate this effect by continuing with open-loop, nominal step parameters after the support exchange has occurred. Even if the capture step did not fully reject the disturbance, we can safely assume that at least some balance was restored and there will be another chance during the next step to correct the remaining instability. By measuring the time that passes between when a step should occur according to the gait signal and when we detect the step using the kinematic model, we determined that our system suffers from a fixed latency of approximately 100ms in the sensorimotor loop. To deal with the latency, we simply subtract a constant offset from the step times that the LIPM controller predicts, before they are used for motion speed estimation.

### IX. EXPERIMENTAL RESULTS

The robustness of our lateral balance controller is best demonstrated in action on a real robot. The accompanying video material [12] shows our robot Dynaped dealing with several disturbances, even during lateral walking. For detailed analysis, we have selected a capture step situation that can be seen in the video at time mark 0:44. Figure 7 shows a sequence of nine steps in total, where the fourth step was a notable capture step as a response to a push from the side. The time line is synchronized with the video. At the beginning of the time line, the robot performs three nominal steps where almost no adaptation takes place. The energy level (bottom) stays nearly constant near the expected value. The instability of the gait frequency estimate (center) can be clearly seen. The lateral push occurs at the end of step number three, approximately at time mark 0:44. The sudden rise of the energy level is clearly visible. In step four, where the value of the noise suppression Gaussian and the energy deviation (bottom) are high at the same time, the adaptation rate rises as well. The result is a smooth transition to a slower step frequency and a larger step size (center). At the same time, the energy deviation is consumed and the adaptation rate drops even before the Gaussian reaches a low value. For the remainder of step four,

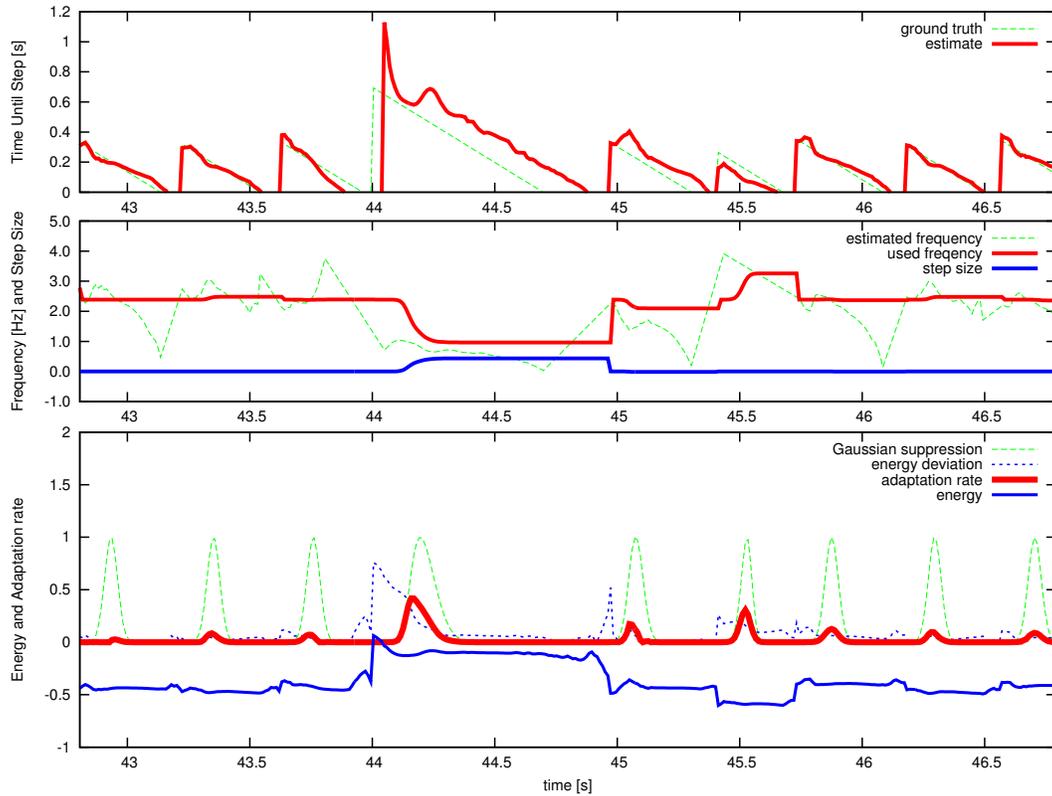


Fig. 7. Data recorded from the robot during a pushing experiment. The top row shows the ground truth and estimated time until support exchange. The middle row shows the actual gait frequency and step size as a result of the adaptation. In the bottom row, the energy curve can be seen and the noise suppression functions that lead to the adaptation rate.

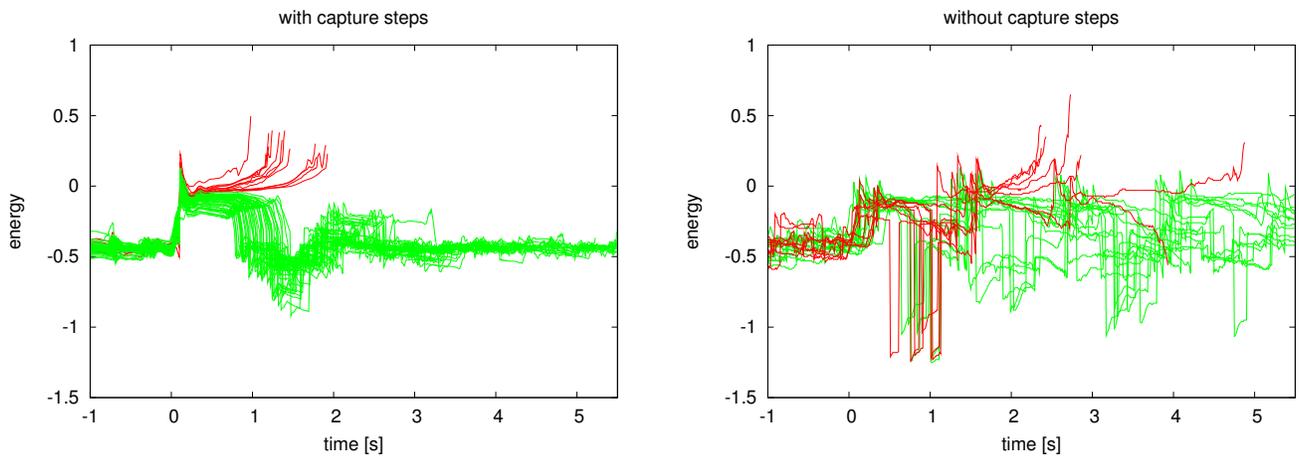


Fig. 8. Time series of measured energy during push experiments with (left) and without (right) capture step feedback. The plotted time series are synchronized so that the disturbance occurs approximately at time mark 0. Non-returning trajectories cannot be captured. These are marked with red color. With capture step feedback, all returning trajectories were successfully stabilized at the nominal energy level after few steps, as can be seen on the left. When the capture steps are turned off, the robot does not return to the nominal energy level, as can be seen on the right. The discontinuities in the energy level without feedback is caused by the robot placing its swing foot on the floor at a wrong time when the body has a high roll angle.

the robot executes the adapted motion speed and step size in open-loop mode. When the support exchange occurs at time mark 0:45, the frequency, the step size and the expected energy level are reset to their default values and a stable step follows, which is close the nominal one. In step number six, there is obviously still some instability left to correct. The robot adapts to a higher frequency for one step and continues with nominal steps starting from step seven. The capture step number four appears to be very effective, as the residual instability of steps five and six can barely be seen with the naked eye when watching the video.

In a quantitative experiment, we evaluated the reliability of the capture step controller by repeatedly pushing the robot from the side during walking on the spot. We applied relatively strong pushes and tried to get close to the non-returning boundary each time. In total, we recorded approximately one hundred pushes. Twenty pushes were performed without lateral capture steps. These are plotted in Figure 8 on the right. The remaining pushes were performed with enabled capture steps. These are plotted in Figure 8 on the left. Twelve pushes were strong enough to produce non-returning trajectories, which we cannot capture. These are explicitly marked with red color. All returning CoM trajectories were successfully recovered. The time series of the energy levels during the pushing experiments are synchronized so that the push always occurs approximately at time mark 0. The effect of the capture steps is very clear to see. On the left, the energy level quickly returns to the nominal level of -0.45 after a few steps. Without capture steps on the right, the robot is unable to recover balance in most cases. We did not produce any non-returning trajectories when the capture step controller was disabled, but the robot still toppled in a number of cases after increasing instability with a badly timed step. These cases are also marked with red color.

We evaluated the computational costs of our algorithm and measured a runtime of 0.06 ms on our reference machine (Intel 2.4 GHz, 32 bit) for the entire cycle starting from Kalman filtering the raw sensor input up to the output of the motion signal. This makes our approach applicable for low power, embedded PCs and even microcontrollers.

## X. CONCLUSIONS

We have presented a bipedal locomotion framework that simplifies the implementation of a closed-loop walk by decomposing the task into hierarchical layers and independent balance components. We implemented balance regulation components for timing and lateral step size and demonstrated their efficiency on a real robot that can cope with any lateral disturbance as long as the robot does not tip over the support leg.

In future work, we are planning to complete the third balance module for the sagittal step size and investigate further improvements of the robustness against external disturbances during walking.

## REFERENCES

- [1] Catherine E. Bauby and Arthur D. Kuo. Active control of lateral balance in human walking. *Journal of Biomechanics*, 33(11):1433–1440, 2000.
- [2] Sven Behnke. Online trajectory generation for omnidirectional biped walking. In *Int. Conf. on Robotics and Automation (ICRA)*, pages 1597–1603, 2006.
- [3] 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.
- [4] J. Ruiz del Solar, R. Palma-Amestoy, R. Marchant, I. Parra-Tsunekawa, and P. Zegers. Learning to fall: Designing low damage fall sequences for humanoid soccer robots. *Robotics and Autonomous Systems Volume*, 57(8), 2009.
- [5] M. Friedmann, J. Kiener, S. Petters, H. Sakamoto, D. Thomas, and O. von Stryk. Versatile, high-quality motions and behavior control of humanoid soccer robots. In *Workshop on Humanoid Soccer Robots of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, pages 9–16, 2006.
- [6] Colin Graf and Thomas Röfer. A closed-loop 3D-LIPM gait for the robocup standard platform league humanoid. In *Workshop on Humanoid Soccer Robots of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, pages 30–37, 2010.
- [7] Kazuo Hirai, Masato Hirose, Yuji Haikawa, and Toru Takenaka. The development of honda humanoid robot. In *Int. Conf. on Robotics and Automation (ICRA)*, 1998.
- [8] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, and K. Yokoi. Biped walking pattern generation by using preview control of zero-moment point. In *Int. Conf. on Robotics and Automation (ICRA)*, pages 1620–1626, 2003.
- [9] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. 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.
- [10] Arthur D. Kuo. Stabilization of lateral motion in passive dynamic walking. *The International Journal of Robotics Research*, 18(9):917–930, 1999.
- [11] 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.
- [12] Marcell Missura and Sven Behnke. Dynaped demonstrates lateral capture steps. <http://www.ais.uni-bonn.de/movies/DynapedLateralCaptureSteps.wmv>.
- [13] 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 *Proc. 2010 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 3150–3156, 2010.
- [14] 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.
- [15] Marc Raibert, Kevin Blankespoor, Gabriel Nelson, Rob Playter, and the BigDog Team. BigDog, the rough-terrain quadruped robot. In *Proceedings of the 17th World Congress*, 2008.
- [16] John Rebula, Fabian Canas, Jerry Pratt, and Ambarish Goswami. Learning capture points for humanoid push recovery. In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2007.
- [17] Koushil Sreenath, Hae won Park, Ioannis Poulakakis, and J. W. Grizzle. Compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on MABEL. *International Journal of Robotics Research*, 2010.
- [18] Benjamin Stephens. Humanoid push recovery. In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2007.
- [19] Ryosuke Tajima, Daisaku Honda, and Keisuke Suga. Fast running experiments involving a humanoid robot. In *Int. Conf. on Robotics and Automation (ICRA)*, 2009.