# Applying Dynamic Walking Control for Biped Robots

Stefan Czarnetzki, Sören Kerner, and Oliver Urbann

Robotics Research Institute
Section Information Technology
Dortmund University of Technology
44221 Dortmund, Germany

**Abstract.** This article presents the application of a novel observer-based control system to achieve reactive motion generation for dynamic biped walking. The proposed approach combines a feedback controller with an online generated feet pattern to assure a stable gait. Experiments in a simulated environment as well as on real robots clearly demonstrate the robustness of the control system. The presented algorithms enable the robot not only to walk dynamically stable but also to cope with major internal disturbances like flaws of the robots internal model and external disturbances like uneven or unstable ground or collisions with objects or other robots.

## 1 Introduction

Humanoid robots are believed to have a high potential for future applications due to the suitability for operation in environments made for humans and due to higher acceptance by people [1], both of which are needed for service and entertainment activities. Despite this, present humanoid robots have a substantial lack of mobility. Even basic tasks such as walking on even ground without an external disturbance are not a trivial challenge. The humanoid shaped form of a two-legged robot results in a high center of mass (CoM) of its body while standing upright. As a result the stance of a humanoid robot is quite unstable, making it likely to tip over. Therefore research on stable biped walking is one of the central problems in this area at the moment. Gait planning for humanoid robots is fundamentally different from the path planning for simple robotic arms. The robots center of mass is in motion all the time while the feet periodically interact with the ground in an unilateral way, meaning that there are only repulsive but no attractive forces between the feet and ground. The movement of the center of mass cannot be controlled directly, but is governed by its momentum and the eventual contact forces arising from ground interaction. These have to be carefully planned in order not to suffer from postural instability.

This paper proposes a control system to achieve reactive motion generation for dynamic biped walking. After giving a brief overview of research on stability aspects of legged robots, the walking pattern generation and control is described. A thorough evaluation is given showing the capability of the system

J. Baltes et al. (Eds.): RoboCup 2009, LNAI 5949, pp. 69–80, 2010.

to generate stable biped walking even under difficult circumstances. The robustness is presented in experiments testing different problem settings as walking with inaccuracies and systematic errors in the model, external disturbances and on uneven or unstable ground.

## 2 Stability

A robot's posture is called balanced and its gait is called statically stable, if the projection of the robot's center of mass on the ground lies within the convex hull of the foot support area (the support polygon). This kind of gait however results in relatively low walking speeds. Similarly natural human gaits are normally not statically stable. Instead they typically consist of phases in which the projection of the center of mass leaves the support polygon, but in which the dynamics and the momentum of the body are used to keep the gait stable. Those gaits are called dynamically stable.

The concept of the zero moment point (ZMP) is useful for understanding dynamic stability and also for monitoring and controlling a walking robot [2]. The ZMP is the point on the ground where the tipping moment acting on the robot, due to gravity and inertia forces, equals zero. The tipping moment is defined as the component of the moment that is tangential to the supporting surface, i.e. the ground. The moment's component perpendicular to the ground may also cause the robot to rotate, but only in a way to change the robot's direction without affecting its stability, and is therefore ignored. For a stable posture, the ZMP has to be inside the support polygon. In the case when it leaves the polygon, the vertical reaction force necessary to keep the robot from tipping over cannot be exerted by the ground any longer, thus causing it to become instable and fall.

In fact, following Vukobratovic's classical notation [3], the ZMP is only defined inside the support polygon. This coincides with the equivalence of this ZMP definition to the center of pressure (CoP), which naturally is not defined outside the boundaries of the robot's foot. If the ZMP is at the support polygon's edge, any additional moment would cause the robot to rotate around that edge, i.e. to tip over. Nevertheless, applying the criteria of zero tipping moment results in a point outside the support polygon in this case. Such a point has been proposed as the foot rotation indicator (FRI) point [4] or the fictitious ZMP (FZMP) [3]. In this so-called fictitious case the distance to the support polygon is an indicator for the magnitude of the unbalanced moment that causes the instability and therefore is a useful measure for controlling the gait.

There are different approaches to generating dynamically stable walking motions for biped robots. One method is the periodical replaying of trajectories for the joint motions recorded in advance, which are then modified during the walk according to sensor measurements [5]. This strategy explicitly divides the problem into subproblems of planning and control. Another method is the realtime generation of trajectories based on the present state of the kinematic system and a given goal of the motion, where planning and control are managed in a unified system. Implementations of this approach differ in the kinematic models being used and the way the sensor feedback is handled. One group requires precise knowledge of the robot's dynamics, mass distribution and inertias of each link to generate motion patterns, mainly relying on the accuracy of the model for motion pattern generation [6,7,8]. A second group uses limited knowledge about a simplified model (total center of mass, total angular momentum, etc.) and relies on feedback control to achieve a stable motion [9,10,11]. The model used for this is often called the inverted pendulum model.

# 3 Motion Generation

This section describes the generation of walking patterns based on a simple inverted pendulum model using a sophisticated preview controller to generate motions resulting in a desired future ZMP movement and to be able to compensate small disturbances or unforeseen forces. The motion generation process can be seen as stages in a pipeline process, which will be described in section 3.1.

## 3.1 Generating the Walking Patterns

The general problem of walking can be seen as an appropriate placement of the feet and a movement of the rest of the body, both of these must satisfy the condition to keep the overall resulting motion stable. The generation of such motion patterns can be divided into separate tasks with one depending on the results of another, therefore forming a pipeline (see figure 1).

The goal of the desired walk is a certain translational and rotational speed of the robot which might change over time, either smoothly i.e. when the robot is slowing down while approaching an object or rapidly i.e. when the robot's high-level objective changes. The translational and rotational speed vector is taken as the input of the motion generation pipeline. This speed vector is the desired speed of the robot, which does not translate to its CoM speed directly for obvious stability reasons, but merely to its desired average. Thus a path is specified that the robot intends to follow. The feet of the robot have to be placed along this path to ensure the correct overall motion of the robot. Alternatively, in scenarios with uneven ground the feet placement at safe positions must be prioritized, resulting in an irregular gait dictating different changes of speed.

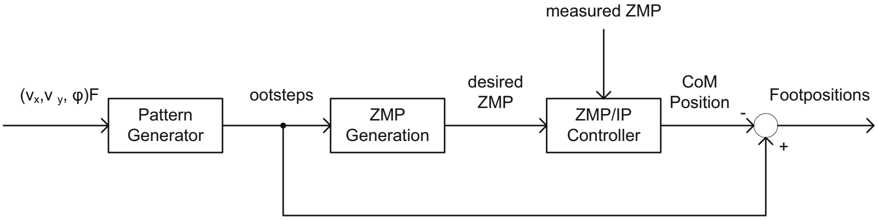


**Fig. 1.** Pipeline visualization of the walking pattern generation process

Once the step patterns are set, these define a region for possible ZMP trajectories to result in stable gaits, namely the support polygon at every given time. A gait can be divided into two phases, a double support phase where both feet are on the ground and a single support phase where only one foot has contact with the ground. During each single support phase the ZMP should be positioned at the center of the ground foot. Consequently in the double support phase the ZMP has to be shifted from one foot to the other. While these restrictions are sufficient to specify the stability of a gait, there is some freedom left in the specification of the exact ZMP trajectory.

The next stage of the process is the generation of a CoM trajectory in which kinematics result in the desired ZMP trajectory. As can be seen later in figures 2(c) and 4 it is not sufficient to shift the CoM at the same time as the ZMP. Instead the CoM has to start moving before the ZMP does. This is realized using a preview control described in more detail in the following section. Its output is a CoM trajectory as shown in figure 1.

All trajectories and positions calculated so far are given in a global world coordinate frame. From the step pattern the feet positions are known, and so is the position of the center of mass at a given time. If the robot's CoM relative to its coordinate frame is known (or assumed to be constant in a simple model), the difference between these directly provides the foot positions in a robot centered coordinate frame. Those can subsequently be transformed into leg joint angles using inverse kinematics.

### 3.2 Modeling Motion Dynamics

The main problem in the process described in section 3.1 is computing the movement of the robot's body to achieve a given ZMP trajectory. For this a simplified model of the robot's dynamics is used, representing the body by its center of mass only. In the single support phase of the walk only one foot has contact with the ground and considering only this contact point and the center of mass, the resulting motion can be described as an inverted pendulum. Its height can be changed by contracting or extending the leg, therefore allowing further control over the CoM trajectory. Restricting the inverted pendulum so that the CoM only moves along an arbitrary defined plane results in simple linear dynamics called the 3D Linear Inverted Pendulum Mode (3D-LIPM) [9]. This plane is given by its normal vector $(k_x, k_y, -1)$ and its intersection with the $z$-axis $z_h$.

For walking on an overall flat terrain the constraint plane is horizontal ($k_x = k_y = 0$) even if the ground itself is uneven. The global coordinate frame depicts the ground as the $x$-$y$-plane and the vertical direction as $z$. Let $m$ be the mass of the pendulum, $g$ the gravity acceleration and $\tau_x$ and $\tau_y$ the torques around the $x$- and $y$-axes, then the pendulum's dynamics are given by

$$\ddot{y} = \frac{g}{z_h} y - \frac{1}{m z_h} \tau_x \tag{1}$$

$$\ddot{x} = \frac{g}{z_h} x + \frac{1}{m z_h} \tau_y . \tag{2}$$

Note that even in the case of a sloped constraint plane the same dynamics can be obtained by applying certain further constraints that are not covered here [9]. According to this model the position $(p_x, p_y)$ of the ZMP on the floor can be easily calculated using

$$p_x = -\frac{\tau_y}{mg} \tag{3}$$

$$p_y = \frac{\tau_x}{mg}. \tag{4}$$

Substituting equations 3 and 4 into 1 and 2 yields the following ZMP equations.

$$p_x = x - \frac{z_h}{g}\ddot{x} \tag{5}$$

$$p_y = y - \frac{z_h}{g}\ddot{y} \tag{6}$$

It can be seen that for a constant height $z_h$ of the constraint plane the ZMP position depends on the position and acceleration of the center of mass on this plane and the $x$- and $y$-components can be addressed separately.

It should be noted for clarification that the ZMP notion of the 3D-LIPM [9] does not take the limitation of the ZMP to an area inside the support polygon into account. Using equations 5 and 6 for planning and controlling stable walking may result in a fictitious ZMP lying outside the support polygon. As mentioned in section 2, this is an indication of an unbalanced moment which causes instability. Since the mathematical notation of the 3D-LIPM used in the following chapters does not involve any distinction based on the support polygon, the general term ZMP will be used hereafter.

### 3.3 Controlling the Motion

Movement of the robot's body to achieve a given ZMP trajectory is thus reduced to planning the CoM trajectory for each direction, resulting in two systems of lesser complexity whose state at any given time is naturally represented by $(x, \dot{x}, \ddot{x})$. The ZMP position $p$ is both the target of the control algorithm and the measurable output of the system. Equations 5 and 6 suggest that the state vector $(x, \dot{x}, p)$ is an equivalent system representation. Choosing this one incorporates the control target into the system state and significantly simplifies further derivations of the controller.

As mentioned previously, the ZMP can not be achieved correctly given its current target value alone, but the CoM needs to start moving prior to the ZMP. Hence the incorporation of some future course of the ZMP is necessary. Such data is available since part of the path to follow is already planned, as described in section 3.1. The following design of a preview controller is described in detail in [12]. It is based on the control algorithms of [13]. [14] already applied some of these to the field of biped walking but used different sensor feedback strategies.

A more natural way of using sensors is presented here. Applying elements common in control theory it is possible to directly incorporate measurements into the system using an observer model as described later in this section. The control algorithm derived here provides the basis for computing a CoM movement resulting in the desired reference ZMP. Assuming the absence of disturbances this system would be sufficient for stable walking.

The system's dynamics can be represented by

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ p \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{g}{z_h} & 0 & -\frac{g}{z_h} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ p \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v \tag{7}$$

where $v = \dot{p}$ is the system input to change the ZMP $p$ according to the planned target ZMP trajectory $p^{ref}$. Discretizing equation 7 with time steps $\Delta t$ yields

$$\mathbf{x}(k+1) = \mathbf{A}_0\mathbf{x}(k) + \mathbf{b}v(k) \tag{8}$$

$$p(k) = \mathbf{c}\mathbf{x}(k) \tag{9}$$

where $\mathbf{x}(k)$ is the discrete state vector $[x\ \dot{x}\ p]^T$ at time $k\Delta t$. Note that $A_0$ describes the system's behavior according to the simplified model. This may not necessarily be identical to the real state transition of the actual robot.

The idea of previewable demand, i.e. the ZMP trajectory due to the planned step pattern, leads to a preview controller [13]. The 3D-LIPM model used to obtain the system dynamics however is only a very simplified approximation of the robot and disturbances and also the state vector itself can not be measured directly in most cases. Therefore it becomes necessary to estimate those from the system input and the measured sensor data in order for the system to work properly under realistic conditions, which leads to the introduction of an observer.The details of the derivation of the control system equations can be found in [12]. The resulting observer for the system is given by equation 10.

$$\widehat{\mathbf{x}}(k+1) = \mathbf{A}_0\widehat{\mathbf{x}}(k) - \mathbf{L}\left[p^{sensor}(k) - \mathbf{c}\widehat{\mathbf{x}}(k)\right] + \mathbf{b}u(k). \tag{10}$$

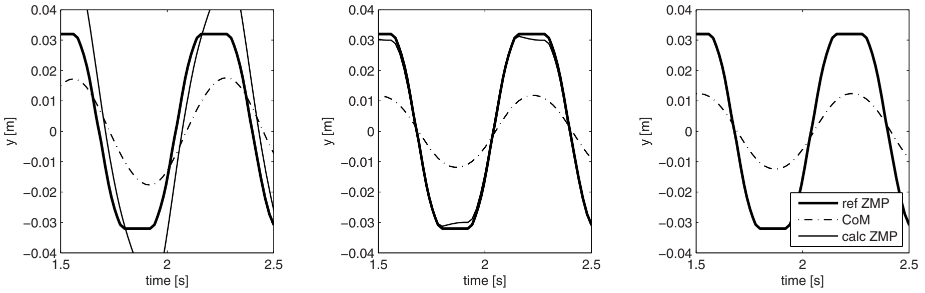
The observer-based controller designed as in equation 11 consists of integral action on the tracking error, proportional state feedback and preview action based on the future demand.

$$u(k) = -G_I \sum_{i=0}^{k} \left[\mathbf{c}\widehat{\mathbf{x}}(i) - p^{ref}(i)\right] - \mathbf{G}_x\widehat{\mathbf{x}}(k) - \sum_{j=1}^{N} G_d(j)p^{ref}(k+j) \tag{11}$$

The gains $G_I$, $\mathbf{G}_x$ and $G_d$ are chosen to optimize the performance index $J$ from

$$J = \sum_{j=1}^{\infty} \left\{ Q_e \left[p(j) - p^{ref}(j)\right]^2 + \Delta\mathbf{x}^T(k)\mathbf{Q}_x\Delta\mathbf{x}(k) + Rv^2(j) \right\} \tag{12}$$

where $\Delta\mathbf{x}$ is the incremental state vector $\Delta\mathbf{x}(k) = \mathbf{x}(k) - \mathbf{x}(k-1)$. The physical interpretation of $J$ is to achieve regulation without an excessive rate of change in the control signal. Both the tracking error and excessive changes in state



(a) N = 10 (b) N = 30 (c) N = 50

**Fig. 2.** Reference ZMP, calculated ZMP and CoM for different numbers $N$ of the previewable values

and control are penalized with weights $Q_e$, $\mathbf{Q}_x$ and $R$, respectively, so that a controller optimizing $J$ achieves a smooth regulation of the controlled system.

The effect of the availability of previewable demand is visualized in figure 2. This issue is directly related to equations 11 and 12 where the infinite horizon spanned by $J$ is approximated by a finite preview window of size $N$. In case of walking, preview windows of the size of a step cycle yield near optimal approximations without adding additional motion delay to that which is already inherent in the motion generation due to step pattern planning.

The resulting control algorithm is visualized in figure 3. An intuitive illustration of this observer-based controller's performance is given in figure 4, where a constant error is added to the ZMP measurement for a period of 1.5 s. This error could be interpreted as an unexpected inclination of the ground or a

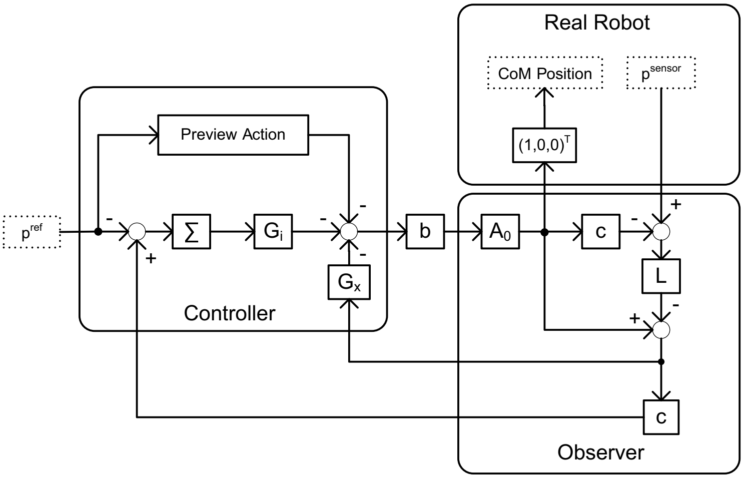


**Fig. 3.** Configuration of the control system

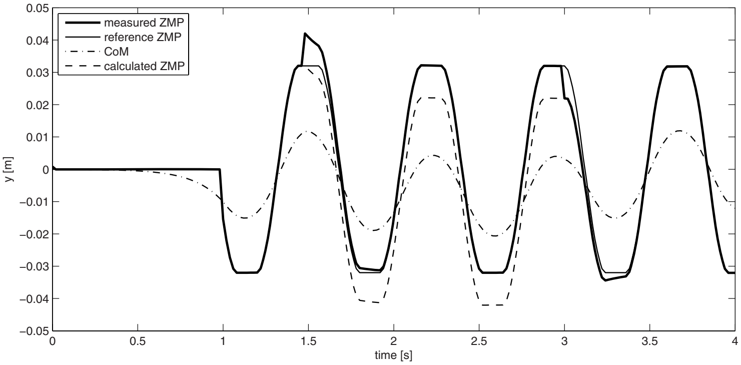


**Fig. 4.** Performance of the controller under the influence of a constant external disturbance resulting in an error in the measured ZMP

constant force pushing the robot to one side. The control system incorporates this difference and compensates by smoothly adjusting the CoM trajectory, which consequently swings more to the opposite direction.

## 4 Evaluation

To demonstrate the benefit of the proposed control system several experiments are evaluated. The scenarios are designed to represent the most common reasons of instability during gait execution, namely external disturbances and deviation from the internal model. External disturbances occur often in the form of discontinuities or irregularities of the ground or external forces caused by collisions. Even with a perfect internal model of the robot, which is nearly impossible to obtain, divergences between real robot occur due to mechanical wearout or permanent external influences. Thus the control algorithm should be capable to level such systematical differences of its model. A video file of the experiments can be found on the homepage of the Robotics Research Institute[1].

The experiments are conducted utilizing the humanoid robot Nao by Aldebaran Robotics. Nao has 21 degrees of freedom, a height of 57 cm, weights 4.5 kg and is equipped with a wide range of extero- and proprioceptive sensors including an accelerometer in its chest. The sensor input and motion output is controlled by a framework running at 50 Hz resulting in discrete time steps $\Delta t$ of 20 ms for the walking control algorithm. The ZMP is measured using the accelerometer.

The first test underlines the need of real world experiments by comparing the differences between walking in simulation and with a real robot. The proposed algorithm is used to generate walking pattern based on the calculated ZMP trajectory. Tests with the help of the general robotics simulator SimRobot [15] using a multi-body model of the robot show that the gait is perfectly stable in

[1] http://www.it.irf.uni-dortmund.de/IT/Robotics/Resource/Application-of-Dynamic-Walking-Control.avi

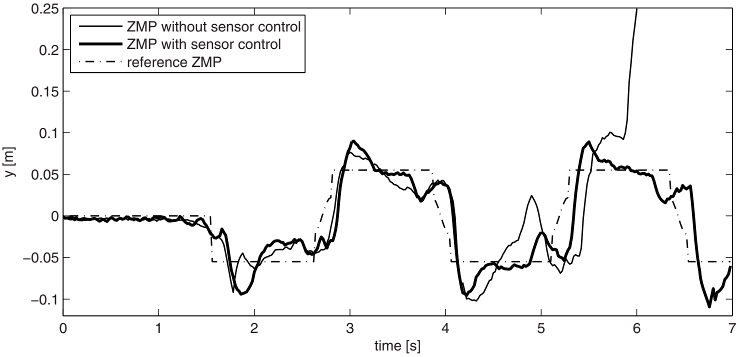


**Fig. 5.** Plot of the ZMP during the model error test

simulation even without sensor feedback. In comparison this gait is executed on the real robot without further adaptations. Figure 5 demonstrates the results of walking straight ahead at speed $v_x = 50\,\frac{\text{mm}}{\text{s}}$ with a step duration of $t_{step} = 2.5\,\text{s}$. As can be seen without sensor control the difference between the reference and the measured ZMP increases over time and the robot starts swinging at second four resulting in a fall around second six. The results of performing the same experiment with feedback control demonstrate the advantage of the proposed closed-loop system. As can be seen in figure 5 the control algorithm is able to adjust the movement according to the flaw of the model and thereby leveling the differences between the ZMPs resulting in a stabilization of the walk. As shown in figure 6 the robot keeps stable even during omnidirectional walking pattern containing substantial changes in speed and directions.

Another design target of the closed-loop system is the capability to level out unforeseen external forces. Hence as an experiment the robot is pushed during a walk at speed $v_x = 50\,\frac{\text{mm}}{\text{s}}$ and a step duration of $t_{step} = 1\,\text{s}$ to simulate the collision with another moving object. Figure 7 illustrates the resulting controller reaction. The collision occurred around second four as can be clearly noticed

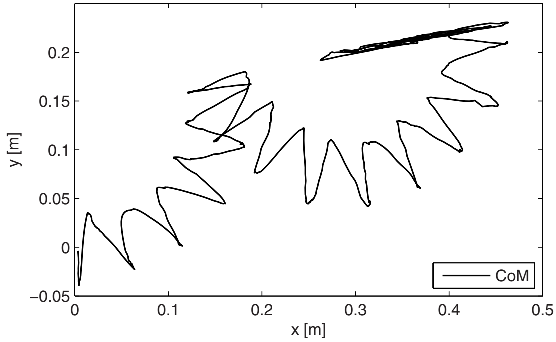


**Fig. 6.** CoM trajectory during omnidirectional walking

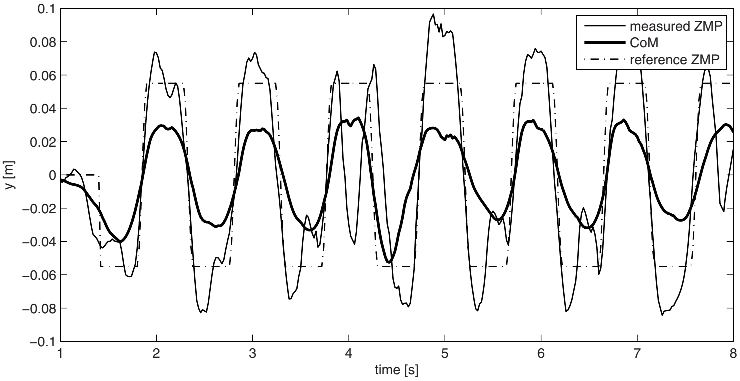
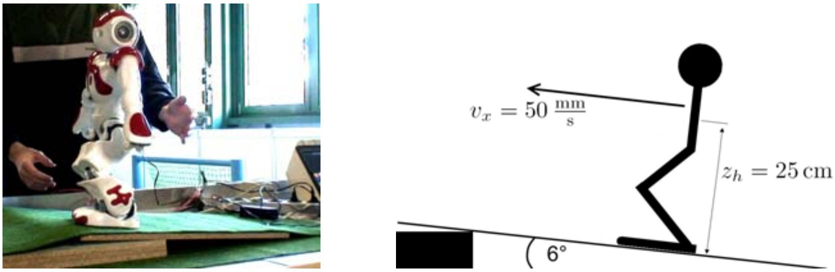


**Fig. 7.** Plot of the ZMP and CoM during the push test



(a) Photo of setup.

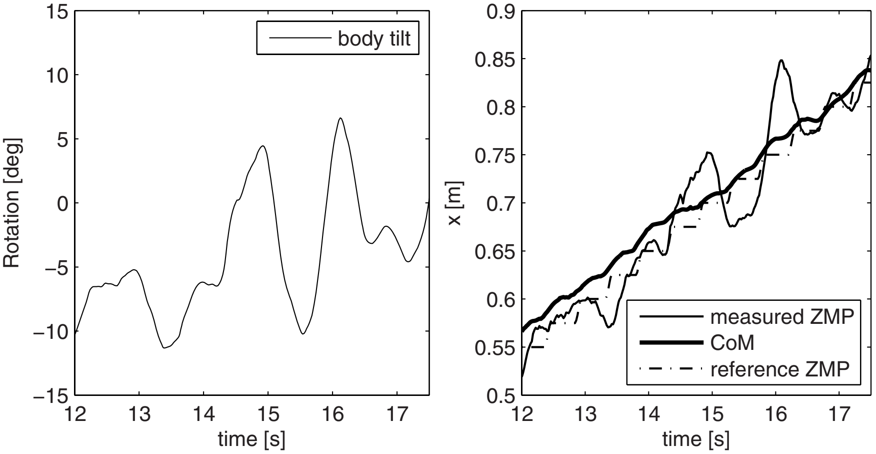
(b) Physical model.

**Fig. 8.** Setup of the uneven floor test

by the abrupt change in the measured ZMP caused by the sudden change of acceleration. The resulting shift of the measured ZMP can be observed in the following second but is compensated by the controller with an adaptation of the desired CoM trajectory stabilizing the walk.

Walking on uneven ground or slopes leads to disturbances often resulting in a fall of the robot. Hence the last experiment, shown in figure 8(b) and 8(a), is designed to simulate these scenarios. Without adaptation the robot is walking up a slope with a gradient angle of 6°. The end of the ramp is not fixed to the block resulting in a rocking movement when walking on top of the edge. Therefore the experiment tests both the capability to level out the continuous error of the slope and to overcome the floor disturbance caused by the rocker.

Figure 9(a) illustrates the measured body orientation. Between second fourteen and fifteen the backward tilt of the robot is disturbed by the tilt of the ramp but becomes stable again in an upright position afterwards. Figure 9(b) shows the ZMP in forwards direction corresponding to this experiment. It can be seen that the robot slips due to the sloped ground shortly after second thirteen



(a) Orientation of Body.

(b) ZMP and CoM.

**Fig. 9.** Results of the uneven floor test

and swings massively when the rocking starts, but reaches a regular walk again towards the end.

The results of the experiments clearly demonstrate the benefit of the proposed sensor feedback control. Without further adaptation to the used hardware or experimental setup the robot is able to adjust its movement to model deviations and external disturbances with the help of sensor feedback in most cases that would clearly result in a fall otherwise.

## 5 Conclusion

This paper presents a novel approach to biped walking based on the online generation of foot trajectories. Special focus is given to the online calculation and control of the CoM movement to achieve the desired ZMP trajectory.

The proposed control system's performance is verified using the humanoid robot Nao of Aldebaran Robotics. The approach generates dynamically stable walking patterns and performs well even under the influence of significant external disturbances. Three experimental setups are chosen as test cases. Each clearly shows the benefit of the observer-based controller for biped walking.

Further improvements can be achieved by introducing a model more complex than the 3D-LIPM. Besides this, the next subject of interest is the integration of different motion patterns other than walking into the control system. These may include planned object contact with maximized impulse, e.g. shooting a ball. A challenge both for motion generation and for perception accuracy is to enable faster and more fluid motions by integrating shooting movements into the normal step patterns.

## References

1. Kanda, T., Ishiguro, H., Imai, M., Ono, T.: Development and evaluation of interactive humanoid robots. Proceedings of the IEEE 92(11), 1839–1850 (2004)
2. Vukobratović, M., Borovac, B., Potkonjak, V.: Towards a unified understanding of basic notions and terms in humanoid robotics. Robotica 25(1), 87–101 (2007)
3. Vukobratovic, M., Borovac, B.: Zero-moment point – Thirty five years of its life. International Journal of Humanoid Robotics 1(1), 157–173 (2004)
4. Goswami, A.: Foot rotation indicator (FRI) point: A new gait planning tooltoevaluate postural stability of biped robots. In: IEEE International Conference on Robotics and Automation, pp. 47–52 (1999)
5. Kim, J.Y., Park, I.W., Oh, J.H.: Experimental realization of dynamic walking of biped humanoid robot khr-2 using zmp feedback and inertial measurement. Advanced Robotics 20(6), 707 (2006)
6. Hirai, K., Hirose, M., Haikawa, Y., Takenaka, T.: The development of honda humanoid robot. In: ICRA, pp. 1321–1326 (1998)
7. Huang, Q., Kajita, S., Koyachi, N., Kaneko, K., Yokoi, K., Arai, H., Komoriya, K., Tanie, K.: A high stability, smooth walking pattern for a biped robot. In: ICRA, p. 65 (1999)
8. Yamaguchi, J., Soga, E., Inoue, S., Takanishi, A.: Development of a bipedal humanoid robot: Control method of whole body cooperative dynamic biped walking. In: ICRA, pp. 368–374 (1999)
9. Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Yokoi, K., Hirukawa, H.: A realtime pattern generator for biped walking. In: ICRA, pp. 31–37. IEEE, Los Alamitos (2002)
10. Raibert, M.: Legged Robots That Balance. MIT Press, Cambridge (2000)
11. Zheng, Y.F., Shen, J.: Gait synthesis for the SD-2 biped robot to climb sloping surface. IEEE Journal of Robotics and Automation 6(1), 86–96 (1990)
12. Czarnetzki, S., Kerner, S., Urbann, O.: Observer-based dynamic walking control for biped robots. Robotics and Autonomous Systems (in press, 2009) (accepted manuscript)
13. Katayama, T., Ohki, T., Inoue, T., Kato, T.: Design of an optimal controller for a discrete-time system subject to previewable demand. International Journal of Control 41(3), 677 (1985)
14. Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Yokoi, K., Hirukawa, H.: Biped walking pattern generator allowing auxiliary zmp control. In: IROS, pp. 2993–2999. IEEE, Los Alamitos (2006)
15. Laue, T., Spiess, K., Röfer, T.: SimRobot - A General Physical Robot Simulator and Its Application in RoboCup. In: Bredenfeld, A., Jacoff, A., Noda, I., Takahashi, Y. (eds.) RoboCup 2005. LNCS (LNAI), vol. 4020, pp. 173–183. Springer, Heidelberg (2006), http://www.springer.de/