Modification of Foot Placement for Balancing Using a Preview Controller Based Humanoid Walking Algorithm

Oliver Urbann and Matthias Hofmann

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

Abstract. Lunges are an important utility to regain balance under strong perturbed biped walking motions. This paper presents a method to calculate the modifications of predefined foot placements with the objective to minimize deviations of the Zero Moment Point from a reference. The modification can be distributed over different points in time to execute smaller lunges, and an arbitrary point in time can be chosen. The calculation is in closed-form, and is embedded into a well-evaluated preview controller with observer based on the 3D Linear Inverted Pendulum Mode (3D-LIPM).

Keywords: 3D-LIPM, ZMP, observer, humanoid robot, reactive stepping, walking algorithm.

1 Introduction

Research in the area of biped robots has become increasingly important over the last years. Especially in environments that contain obstacles and barriers such as stairs, or small objects lying on the ground, biped robots are advantageous in comparison to wheeled robots. Bipedal locomotion in such surroundings is naturally susceptible to disturbances that may occur when the robot collides with those objects. As a precedent step, biped robots must be capable to regain balance even on flat grounds to prevent from damage.

1.1 Related Work

A widely used criterion to determine the stability of a humanoid robot is the Zero Moment Point (ZMP). It was invented by Miomir Vukobratović [1] based on the term support polygon which is the convex hull of all contact points of the feet with the ground. If the ZMP is inside the support polygon, the ground does not exert torques around the roll and pitch axes of the feet, and thus the robot can be considered as stable.

The ZMP can additionally be exploited to define the locomotion of the robot. For this purpose, the reference ZMP is chosen to stay always inside the support

S. Behnke et al. (Eds.): RoboCup 2013, LNAI 8371, pp. 420–431, 2014.

polygons which result from the planned foot positions. As this approach implies a computational solution of non-linear equations which are runtime-intensive, Kajita et al. introduced a linear model of the robot that associates the movement of the center of mass (CoM) of the robot to the ZMP, the 3D linear inverted pendulum mode (3D-LIPM) [2].

There are many approaches to derive a CoM trajectory from the desired ZMP employing the 3D-LIPM, e.g. using a convolution sum [3], or an analytical solution [4]. Kajita et al. proposed a method for an arbitrary ZMP trajectory based on linear-quadratic regulator [5]. To incorporate measurements from different sensors, an observer can be integrated [6,7,8]. While a stabilizing effect of this approach can be shown[1] [8], it is limited due to the fact that major disturbances cannot be compensated without a lunge[2].

In the literature, there are various approaches to determine a step that fits best to the desired motion and dynamical state of the robot. A reasonable way for this is to take the orbital energy into account. Doing so, the switching time of the stance foot can be solved analytically [9]. While this is in fact not a lunge where the desired foot position is modified, Urata et al. proposed a technique based on a preview controller that optimizes the foot placement numerically [10]. Other approaches are based on a concept called Capture Point (CP) [11] which is defined as the foot position where the robot comes to a full stop with the CoM directly above. The CP can be calculated using an extension of the 3D-LIPM, and this can be used to generate a walking motion [12]. While the steps are calculated in a closed form, a major drawback is the validity of the CP which is not constant over time [13]. Moreover, the CP approach does not consider the point in time when the stance foot is changed. This both drawbacks are the motivation to invent the foot placement estimator (FPE) and the foot placement indicator (FPI) [14]. However, all points (CP, FPE and FPI) have the drawback in common that generating a walking motion using these points is counter-intuitive as they were originally introduced to immediately stop the robot after disturbances like a push.

Another approach to simultaneously plan the motion and the desired foot position according to the state of the robot including disturbances is to formulate it as a quadratic program (QP) solved using Model Predictive Control (MPC) [15]. Also Morisawa et al. use a quadratic program to find appropriate foot placements [16]. However, both solutions can reveal large CoM and ZMP deviations under certain circumstances. In these cases, the disturbances should be balanced by other means than stepping [17]. Furthermore, the lunge can only be calculated by numerical optimization in both approaches.

[1] A video of the closed-loop performance can be watched here:
`https://www.youtube.com/watch?v=ZuebspajdZU`

[2] The term for a step that is needed to balance disturbances that cannot be balanced in another way. Normally a lunge is a modification of a planned foot position. In the literature also referred to as reactive stepping or just stepping.

1.2 Objectives

This paper proposes a method to modify the predefined foot placement in order to regain balance after disturbances were observed. To this end, the following required properties of the system are defined:

1. The system should deal with a reference ZMP of an arbitrary shape to avoid oscillations of the body. E.g. a constant ZMP speed in walking direction ensures a constant body speed.
2. The system should ensure that the actual position of the feet is equal to the desired if needed. In case of disturbances, instabilities must then be balanced conventionally. Otherwise it would be difficult to stop next to an object without a collision.
3. The system must always provide and guarantee a solution for the modification of the foot placement.
4. As the controller is supposed to run on platforms with limited computing power, a calculation in closed-form is needed.
5. Because kinematical constraints must be considered to avoid self-collision, it is important to be able to select a random point in time for the lunge execution.
6. To avoid large steps, it should be possible to distribute a single lunge over multiple steps.

Since a reference ZMP of an arbitrary shape is desired, the preview controller with observer approach as presented in [6,7,8] is the basis for this work.

1.3 Structure

The remainder of the paper is structured as follows: Section 2 briefly summarizes the main concepts of the used preview controller and observer, and reflects the output of the balancing system in case of disturbances. Based on this, section 3 presents a method to suppress these disturbances and outlines how the modification of the foot placement is processed. The approach is evaluated accompanied by some examples in section 4 by using a simulation environment. The paper concludes in section 5 including future work. Formulas and algorithms presented in this paper can be downloaded as described in section 6.

2 Preview Controller and Observer

This section describes the basic concepts that are used in the preview controller. Afterwards an observer is presented that is utilized to incorporate sensor measurements. Please note that all derivations are for the sagittal plane but also hold for the coronal plane. The final walking algorithm therefore consists of two controllers, one for each plane, working in parallel. For details about the overall algorithm see [8].

A fast way to determine the target position of the CoM is a preview controller as proposed by Kajita et al. [5]. It is a linear-quadratic regulator (LQR) that

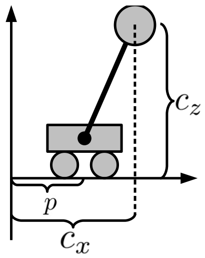

Fig. 1. Cart model of a linear inverted pendulum

requires a preview of the reference ZMP, denoted here as a vector $\boldsymbol{p}^{ref}$ of all reference points for the walk. A detailed description is given in [7], and proofs can be found in [18].

The LQR requires a linear model of the robot. Hence, the 3D-LIPM which abstracts the robot to a single point mass is used to construct the controller. This mass is supported by a single pole, and can only move in a plane with a constant height over ground (see Figure 1). To keep the mass at the height c_z, the ZMP p must be at the x position of the pivot joint. The acceleration of the CoM in x direction $\ddot{c_x}$ must then satisfy:

$$p = c_x - \frac{c_z}{g}\ddot{c_x}, \tag{1}$$

where g is the gravity.

A discrete-time system (with a sampling frequency of $\frac{1}{\Delta t}$) describes the physical behavior, and can be formulated as:

$$\mathbf{x}(k+1) = \mathbf{A}_0\mathbf{x}(k) + \mathbf{b}u(k) \tag{2}$$

$$p(k) = \mathbf{C}\mathbf{x}(k), \mathbf{b} = (0, 0, \Delta t)^T \tag{3}$$

where $u(k)$ denotes the controller output which is the time derivative of the ZMP $\dot{p}$, and $\mathbf{x}(k) = (c_x(k), \dot{c}_x(k), p(k))^T$ is the state of the system. Matrix $\mathbf{C} = (0, 0, 1)$ selects the ZMP as the system output and therefore as the reference. The matrix

$$\mathbf{A}_0 = \begin{bmatrix} 1 & \Delta t & 0 \\ \frac{g}{c_z}\Delta t & 1 & -\frac{g}{c_z}\Delta t \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

incorporates the 3D-LIPM equation 1. To derive the controller $u(k)$ in equation 2, a performance index is defined that meets the following requirements: First, the expected ZMP should be equal to the reference. Second, the resulting trajectory of the CoM has to be smooth. Third, the output $\dot{p}$ of the controller should be low as long as the reference is tracked well. Thus:

$$J = \sum_{j=0}^{\infty} \left\{ Q_e \left[p(j) - \boldsymbol{p}_j^{ref} \right]^2 + \Delta\mathbf{x}^T(j) Q_x \Delta\mathbf{x}(j) + R\Delta u^2(j) \right\}, \tag{5}$$

where $\Delta\mathbf{x}$ and Δu are the incremental state vector and controller output respectively. The gains R, Q_x, Q_e are platform-dependent parameters and $\boldsymbol{p}^{ref}$ is the preview of the desired ZMP with a length of $N+1$.

The controller minimizing J is given by:

$$u(k) = -G_I \sum_{i=0}^{k} \left[\mathbf{C}\mathbf{x}(i) - \boldsymbol{p}_i^{ref} \right] - \mathbf{G}_x \mathbf{x}(k) - \sum_{j=0}^{N} G_d(j) \boldsymbol{p}_{k+j}^{ref}, \tag{6}$$

where the gains can be calculated as follows:

$$\mathbf{B} = \begin{bmatrix} \mathbf{Cb} \\ \mathbf{b} \end{bmatrix}, \mathbf{I} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \mathbf{F} = \begin{bmatrix} \mathbf{CA}_0 \\ \mathbf{A}_0 \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} Q_e & 0 \\ 0 & \mathbf{C}^T Q_x \mathbf{C} \end{bmatrix}, \mathbf{A} = [\mathbf{I}, \mathbf{F}] \tag{7}$$

$$G_I = \left[R + \mathbf{B}^T \mathbf{P} \mathbf{B} \right]^{-1} \mathbf{B}^T \mathbf{P} \mathbf{I}, \quad \mathbf{G}_x = \left[R + \mathbf{B}^T \mathbf{P} \mathbf{B} \right]^{-1} \mathbf{B}^T \mathbf{P} \mathbf{F} \tag{8}$$

$$G_d(j) = -\left[R + \mathbf{B}^T \mathbf{P} \mathbf{B} \right]^{-1} \mathbf{B}^T \left[\mathbf{A}_c^T \right]^j \mathbf{P} \mathbf{I}, j = 0, 1, ..., N. \tag{9}$$

The closed-loop matrix $\mathbf{A}_c$ is defined as

$$\mathbf{A}_c = \mathbf{A} - \mathbf{B} \left[R + \mathbf{B}^T \mathbf{P} \mathbf{B} \right]^{-1} \mathbf{B}^T \mathbf{P} \mathbf{A}, \tag{10}$$

where $\mathbf{P}$ is the solution of the discrete-time matrix Riccati equation that can be solved offline before walking:

$$\mathbf{P} = \mathbf{A}^T \mathbf{P} \mathbf{A} - \mathbf{A}^T \mathbf{P} \mathbf{B} \left[R + \mathbf{B}^T \mathbf{P} \mathbf{B} \right]^{-1} \mathbf{B}^T \mathbf{P} \mathbf{A} + \mathbf{Q}. \tag{11}$$

In this paper it is assumed that the actual ZMP and the current CoM position can be measured [8], e.g. by making use of the Force Sensitive Resistors (FSR) in the feet alongside with measured joint angles, or the attitude of the body. Hence, Matrix $\mathbf{C}_m$ is defined as

$$\mathbf{C}_m = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{12}$$

which is used to select the measurable part $\mathbf{y}$ of the state vector $\mathbf{x}$:

$$\mathbf{y}(k) = \mathbf{C}_m \cdot \mathbf{x}(k) \hat{=} \begin{bmatrix} c_x^{snr}(k) \\ p^{snr}(k) \end{bmatrix}. \tag{13}$$

The discrete time system with sensor feedback is defined slightly different from the open-loop case:

$$\widehat{\mathbf{x}}(k+1) = \mathbf{A}_0 \widehat{\mathbf{x}}(k) + \mathbf{L} \left[\mathbf{y}(k) - \mathbf{C}_m \widehat{\mathbf{x}}(k) \right] + \mathbf{b} \widehat{u}(k). \tag{14}$$

Equation 14 incorporates the measurements by adding $\boldsymbol{e}(k) := \mathbf{L} \left[\mathbf{y}(k) - \mathbf{C}_m \widehat{\mathbf{x}}(k) \right]$ to equation 2: The summand expresses the difference between the estimated state $\widehat{\mathbf{x}}$ and the measurements $\boldsymbol{y}$ multiplied by a matrix $\mathbf{L}$ that is constructed by executing the LQR procedure again. The latter task can be performed with built-in functions of Matlab[3] or Octave[4]. Figure 2 illustrates an example walk where an error is balanced by the controller. In the left part of the figure an error of $\boldsymbol{\delta}(1.77) := \mathbf{y} - \mathbf{C}_m \widehat{\mathbf{x}} = (0.005\,\mathrm{m}, 0\,\mathrm{m})^T$ (a measured error in the CoM position) is added at time $t \approx 1.8\,\mathrm{s}$. One frame after the error was measured, the observed

[3] `http://www.mathworks.com`

[4] `http://www.gnu.org/software/octave/`

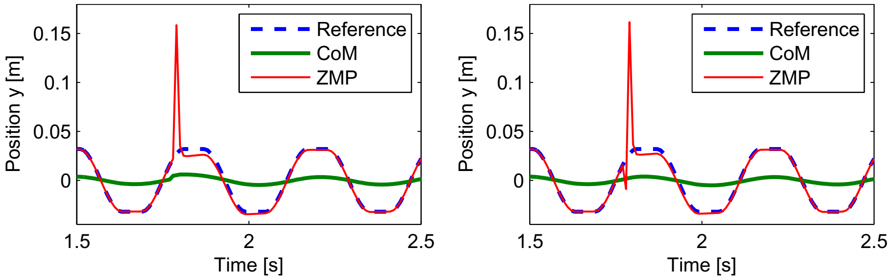

Fig. 2. Lateral plot of disturbed walk balanced by the controller. "Reference" denotes the reference ZMP $\hat{\boldsymbol{p}}^{ref}$, "CoM" the observed CoM position $\hat{c}_x$ and "ZMP" the observed ZMP position $\hat{p}$. On the left, a CoM position error is induced, on the right a ZMP error.

ZMP reveals a large deviation. This is the reaction of the controller that accelerates the CoM towards the CoM trajectory that is optimal under the given reference ZMP. Starting with the next frame, the CoM is then decelerated until the CoM position and velocity reaches optimal values. This deceleration yields to a ZMP error into the opposite direction.

On the right a ZMP error of $\boldsymbol{\delta}(1.77) = (0\,\text{m}, -0.05\,\text{m})^T$ is added. As can be seen, a similar reaction of the controller is the result.

3 Integration of Foot Placement Modification

The behavior of the controller presented in the previous section is not optimal and leads to further deviations. Under circumstances this can result in a ZMP outside the support polygon and a fall down. The central idea presented in this paper is the modification of the future foot positions, and consequently the reference ZMP by carrying out a lunge instead of forcing back the erroneous CoM to the original trajectory. A positive side effect is that the additional ZMP deviation due to the subsequent deceleration after the large acceleration also disappears. In order to calculate the modification, we postulate that the expected ZMP of the closed-loop system is equal to an open-loop system. The latter is not disturbed (and therefore reveals no ZMP deviation) at the time of the first reaction of the controller. The closed-loop system then also does not reveal ZMP deviations.

The case where a ZMP error is measured must be handled slightly different. In this case, a significant larger error is added to the CoM velocity since the CoM acceleration is directly related to the ZMP by the 3D-LIPM equation. If this is ignored, it would result in a similar deceleration deviation as can be seen in figure 2. If the cancellation of any ZMP deviation is desired as soon as possible, an opposite ZMP deviation must be applied in the frame after the error occured to instantaneously decelerate the CoM.

The described deliberations lead to the following requirement:

$$\boldsymbol{C}\widehat{\mathbf{x}}(T+2) - \boldsymbol{C}\mathbf{x}(T+2) \stackrel{!}{=} -\boldsymbol{C}\boldsymbol{e}(T), \tag{15}$$

where T is the time when the error is measured. Please note that $T+1$ is the time when the error appears in the state vector. The controller $\widehat{u}(T+1)$ then induces the ZMP deviation into $\widehat{\mathbf{x}}(T+2)$.

In order to comply with the system requirements defined in section 1.2, the objective is to find a map S calculating modifications in closed-form fulfilling equation 15:

$$S : \{t_0, \ldots, N\} \times \mathbb{R}^2 \to \mathbb{R}, (t, \boldsymbol{e}) \mapsto m \tag{16}$$

where $t \geqslant t_0$ is the point in time where the adjustment m of the reference ZMP is commenced (relative to the point in time where $\boldsymbol{e}$ is measured). The application of the scalar value m can be derived by observing human beings performing a lunge. It is a modification applied to the position of all foot steps starting at t. For this purpose, a ZMP modificator $\boldsymbol{p}^m$ that can be added to the reference ZMP is defined as:

$$\forall k < T + t : \boldsymbol{p}_k^m = 0 \tag{17}$$

$$\forall k \geq T + t : \boldsymbol{p}_k^m = m, \tag{18}$$

Consequently, the modified reference ZMP is:

$$\hat{\boldsymbol{p}}^{ref} = \boldsymbol{p}^{ref} + \boldsymbol{p}^m. \tag{19}$$

Obviously, the first possible adaption time t_0 of the reference ZMP is when the disturbance occurs in the state vector at $T+1$. However, equation 15 implicitly includes the reference ZMP $\hat{\boldsymbol{p}}_{T+2}^{ref}$ but assumes $\boldsymbol{p}_{T+2}^m = 0$. Therefore the adjustment can start at $t_0 = 3$.

To derive S, we first examine the influence of the measured error on the state vector at $T+2$ by recursively applying equation 14 and 6 :

$$\boldsymbol{C}\widehat{\mathbf{x}}(T+2) = \boldsymbol{C}\left(\mathbf{A}_0\left(\mathbf{A}_0\widehat{\mathbf{x}}(T) + \boldsymbol{e}(T) + \mathbf{b}\widehat{u}(T)\right) + \boldsymbol{e}(T+1) + \mathbf{b}\widehat{u}(T+1)\right) \tag{20}$$

$$\widehat{u}(T+1) = \widehat{I}(T+1) + \widehat{X}(T+1) + \widehat{D}\left(\left(\hat{\boldsymbol{p}}_{T+1}^{ref}, \ldots, \hat{\boldsymbol{p}}_{T+1+N}^{ref}\right)\right) \tag{21}$$

$$\widehat{I}(T+1) = -G_I \sum_{i=0}^{T+1} \left[\boldsymbol{C}\left(\mathbf{A}_0\widehat{\mathbf{x}}(i-1) + \boldsymbol{e}(i-1) + \boldsymbol{b}\widehat{u}(i-1)\right) - \hat{\boldsymbol{p}}_i^{ref}\right] \tag{22}$$

$$\widehat{X}(T+1) = -\boldsymbol{G}_x\left(\mathbf{A}_0\widehat{\boldsymbol{x}}(T) + \boldsymbol{e}(T) + \boldsymbol{b}\widehat{u}(T)\right) \tag{23}$$

$$\widehat{D}(\hat{\boldsymbol{p}}^{ref}) = -\sum_{i=0}^{N}\left[\boldsymbol{G}_d(i)\hat{\boldsymbol{p}}_i^{ref}\right] = -\sum_{i=0}^{N}\left[\boldsymbol{G}_d(i)\boldsymbol{p}_i^{ref}\right] - \sum_{i=0}^{N}\left[\boldsymbol{G}_d(i)\boldsymbol{p}_i^{m}\right] \tag{24}$$

$$= -\sum_{i=0}^{N}\left[\boldsymbol{G}_d(i)\boldsymbol{p}_i^{ref}\right] - m\sum_{i=t}^{N}\boldsymbol{G}_d(i), \tag{25}$$

From 24 to 25, equations 17 and 18 are applied. Since the modification is constant starting at t, the sum of the mapping $\boldsymbol{p}^m$ can be replaced by the corresponding scalar value m.

Equations 20-25 are simplyfied by exploiting the following statements:

$$\forall k \leq T : \widehat{\mathbf{x}}(k) = \mathbf{x}(k), \widehat{u}(k) = u(k) \tag{26}$$

$$\forall k \neq T : \boldsymbol{e}(k) = 0 \tag{27}$$

$$\forall k \leq T + t, 2 < t : \hat{\boldsymbol{p}}_k^{ref} = \boldsymbol{p}_k^{ref}, \tag{28}$$

Equations 27 and 26 hold since this derivation is based on an impulse like error at time T. It is important to recognize that no disturbance separates the open-loop case from the closed-loop case before T. Hence, equations 20-25 are similar in the open-loop case except for the error terms $\boldsymbol{e}$ and the ZMP modification $\boldsymbol{p}^m$. On the left side of equation 15 this cancels out every term that does not contain $\boldsymbol{e}$ or m:

$$-\boldsymbol{C}\boldsymbol{e}(T) = \boldsymbol{C}\widehat{\mathbf{x}}(T+2) - \boldsymbol{C}\mathbf{x}(T+2) \tag{29}$$

$$= \boldsymbol{C}\left(\boldsymbol{A}_0\boldsymbol{e}(T) + \boldsymbol{b}\left(-\boldsymbol{G}_I\boldsymbol{C}\boldsymbol{e}(T) - \boldsymbol{G}_x\boldsymbol{e}(T) - m\sum_{i=t}^{N}\boldsymbol{G}_d(i)\right)\right) \tag{30}$$

$$\Leftrightarrow m = \underbrace{\left(\boldsymbol{C}\boldsymbol{b}\sum_{i=t}^{N}\boldsymbol{G}_d(i)\right)^{-1} \boldsymbol{C}\left(\boldsymbol{A}_0 + \boldsymbol{I}_{3\times 3} - \boldsymbol{b}\boldsymbol{G}_I\boldsymbol{C} - \boldsymbol{b}\boldsymbol{G}_x\right)}_{=:\boldsymbol{G}_e(t)}\boldsymbol{e}(T). \tag{31}$$

Function S can now be defined as:

$$S(t, \boldsymbol{e}) = \boldsymbol{G}_e(t)\boldsymbol{e}, \quad t \in \{3, \ldots, N\}. \tag{32}$$

Therefore the modification for the reference ZMP and the foot steps can be calculated by a simple multiplication. The gains $\boldsymbol{G}_e$ can be calculated in closed-form together with the gains described in section 2 offline before walking.

In some cases it is not possible to apply only one lunge to balance error $\boldsymbol{e}$, e.g. if the sidestep would be too large for the given leg length. Therefore it is possible to split the modification of a single step to n modifiers that can be applied at any desired point. To do so, the error has to be distributed over different smaller modifications according to a desired ratio: $\boldsymbol{e} = \alpha_1\boldsymbol{e} + \ldots + \alpha_n\boldsymbol{e}$ with $\alpha_1 + \ldots + \alpha_n = 1$. The modifiers $m_1, \ldots, m_n$ can then be computed by $m_k = S(t_k, \alpha_k\boldsymbol{e})$.

To show that this kind of system has the same stabilizing effect, its influence to the controller must be examined. As can be seen in the controller equation 6, the only term incorporating the future reference ZMP is $\widehat{D}$ as formulated in equation 24. The error can be split between different lunges since $\widehat{D}$ is a linear map (as shown in equations 24 and 25). The resulting modifiers can be applied at random future points in time since S calculates the modifier for the arbitrary time t by definition.

4 Examples and Evaluation

This section starts with an example showing the stabilizing effect on a robot using the 3D-LIPM. It depicts the input ($\hat{\boldsymbol{p}}^{ref}$) and output ($c_x$ and p) of a walk that

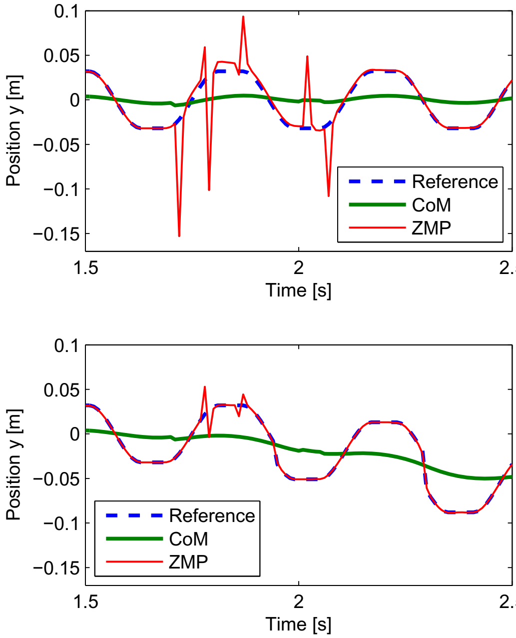

Fig. 3. Example walk with measured errors. "Reference" denotes the reference ZMP $\hat{\boldsymbol{p}}^{ref}$, "CoM" the observed CoM position $\hat{c}_x$ and "ZMP" the observed ZMP position $\hat{p}$. In the top graph the errors are balanced by the controller only, in the bottom graph lunges are included.

is disturbed by different impulse-like errors. As shown in Figure 3, the following errors are measured: $\boldsymbol{\delta}(1.7) = (-0.005, 0)^T, \boldsymbol{\delta}(1.77) = (0, 0.05)^T, \boldsymbol{\delta}(1.85) = (0, -0.02)^T, \boldsymbol{\delta}(2) = (0.003, 0)^T$ and $\boldsymbol{\delta}(2.05) = (-0.003, 0)^T$. In the upper graph, the controller balances the disturbance by forcing the CoM back to the initial movement without a modification of the foot positions.

The lower graph displays an example where lunges are planned for the double support phases at 1.95 s and 2.3 s. In this example, we define that 80% of a measured error has to be corrected by the first lunge, and 20% by the second under the assumption that the error occurs before the first lunge. The later errors are compensated by the second lunge exclusively. As can be seen, all measured CoM errors (at 1.7 s, 2 s and 2.05 s) can be balanced without any ZMP disturbance. The measured ZMP errors at 1.77 s and 1.85 s exhibit the intended ZMP deviations as desired in equation 15 within the subsequent frame. Nevertheless, even with the desired deviations the errors are balanced with far less ZMP deviations compared to the case without modifications.

However, the requirement in equation 15 is based on the 3D-LIPM. Since this is a strong abstraction, the proposed approach has to be evaluated on more complex models. It is also unclear if the inverted ZMP deviation at $T+1$ ($-\boldsymbol{Ce}(T)$ in

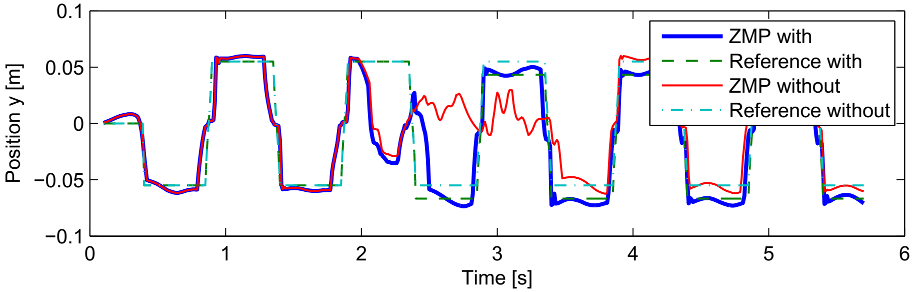

Fig. 4. Multibody simulation of a disturbed walk comparing closed-loop balancing with modification of the foot position ("ZMP/Reference with") with balancing without modification ("ZMP/Reference without")

equation 15) has the desired effect on a real robot. Therefore a walking algorithm incorporating the results of section 2 and 3 is evaluated using SimRobot [19] with a model of the Nao by Aldebaran Robotics[5]. The implementation used for this experiments executes lunges only for larger disturbances and only for the next step since later the lunge would be significantly larger.

In this experiment, the robot walks at a speed of $5 \frac{\mathrm{cm}}{s}$ with a step duration of 1 s. As can be seen in Figure 4, at time ~ 2 s a force of 1000 N is applied for 0.01 s. This leads to a disturbance that is compensated by the system after 3 additional steps without carrying out a lunge. When a lunge is executed, stabilization can be achieved in the next step.

It has to be emphasized that results of experiments using a multibody simulation or a real robot strongly depend on the implementation of the overall walking algorithm. For example, the algorithm must not move the swinging foot any further when it accidentally hits the ground caused by tipping. The swinging foot must also be in parallel to the ground at that moment. However, a more detailed evaluation of those algorithmic details is out of the scope of this paper.

5 Conclusion and Future Works

This paper depicts the drawbacks of balancing without modification of footsteps. It can lead to further ZMP deviations besides the measured errors. To overcome this, a requirement for the controller output is formulated and rearranged to a set of gains that can be utilized to calculate the modification in closed-form. This closed-form calculation fulfils objectives 2 (by simply not applying any modification), 3 and 4 (since function S can be calculated by a simple matrix multiplication). Objective 1 is fulfilled by utilizing a preview controller and objectives 5 and 6 are fulfilled since function S can be applied on fractions of the measured error at a random point in time (see end of section 3).

[5] `http://www.aldebaran-robotics.com`

As shown in the example, balancing by performing lunges results into a walk with far less ZMP errors. It can also help stabilising a walk that relies on a more complex model of the robot.

However, this strongly depends on the realization of the entire walking algorithm. The implementation of an adequate walking algorithm that exploits the advantages of the proposed method to stabilize the walk of a real robot is of particular importance for future work. Moreover, the current requirement of the immediate correction of the CoM speed to ensure a correct ZMP leads to the term $-\boldsymbol{Ce}(T)$ in equation 15. While this still results in far less ZMP errors, it would be beneficial to recover the desired speed by a different modification of the reference ZMP. First studies show that it is possible to avoid any ZMP error till an arbitrary point in time, even in presence of false CoM speeds. Currently, this is only possible by applying the modifications in every succeeding frame.

6 Supplementary Material

The algorithms and formulas presented in this paper can be downloaded as Matlab/Octave scripts: `http://irf.tu-dortmund.de/nao-devils/download/ZMPIPController2012.zip`. The examples shown in Figure 2 and 3 are also included.

References

1. Vukobratović, M., Borovac, B.: Zero-moment point – Thirty five years of its life. International Journal of Humanoid Robotics 1(1), 157–173 (2004)
2. Kajita, S., Kanehiro, F., Kaneko, K., Yokoi, K., Hirukawa, H.: The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. In: Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, pp. 239–246 (2001)
3. Kim, J.H.: Walking pattern generation of a biped walking robot using convolution sum. In: 2007 7th IEEE-RAS International Conference on Humanoid Robots, November 29-December 1, pp. 539–544 (2007)
4. Harada, K., Kajita, S., Kaneko, K., Hirukawa, H.: An analytical method for real-time gait planning for humanoid robots. International Journal of Humanoid Robotics 3(01), 1–19 (2006)
5. Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., Hirukawa, H.: Biped walking pattern generation by using preview control of zero-moment point. In: ICRA, pp. 1620–1626. IEEE (2003)
6. Czarnetzki, S., Kerner, S., Urbann, O.: Applying dynamic walking control for biped robots. In: Baltes, J., Lagoudakis, M.G., Naruse, T., Ghidary, S.S. (eds.) RoboCup 2009. LNCS (LNAI), vol. 5949, pp. 69–80. Springer, Heidelberg (2010)
7. Czarnetzki, S., Kerner, S., Urbann, O.: Observer-based dynamic walking control for biped robots. Robotics and Autonomous Systems 57(8), 839–845 (2009)
8. Urbann, O., Tasse, S.: Observer based biped walking control, a sensor fusion approach. Autonomous Robots, 1–13 (2013)

9. Alcaraz-Jiménez, J.J., Missura, M., Martínez-Barberá, H., Behnke, S.: Lateral disturbance rejection for the nao robot. In: Chen, X., Stone, P., Sucar, L.E., van der Zant, T. (eds.) RoboCup 2012. LNCS (LNAI), vol. 7500, pp. 1–12. Springer, Heidelberg (2013)
10. Urata, J., Nshiwaki, K., Nakanishi, Y., Okada, K., Kagami, S., Inaba, M.: Online decision of foot placement using singular lq preview regulation. In: 2011 11th IEEE-RAS International Conference on Humanoid Robots (Humanoids), pp. 13–18 (October 2011)
11. Pratt, J.E., Carff, J., Drakunov, S.V., Goswami, A.: Capture point: A step toward humanoid push recovery. In: Humanoids, pp. 200–207. IEEE (2006)
12. Englsberger, J., Ott, C., Roa, M., Albu-Schaffer, A., Hirzinger, G.: Bipedal walking control based on capture point dynamics. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4420–4427 (September 2011)
13. Yun, S.K., Goswami, A.: Momentum-based reactive stepping controller on level and non-level ground for humanoid robot push recovery. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3943–3950 (September 2011)
14. van Zutven, P., Kostic, D., Nijmeijer, H.: Foot placement for planar bipeds with point feet. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), pp. 983–988 (May 2012)
15. Herdt, A., Diedam, H., Wieber, P.B., Dimitrov, D., Mombaur, K., Diehl, M.: Online walking motion generation with automatic footstep placement. Advanced Robotics 24(5-6), 719–737 (2010)
16. Morisawa, M., Harada, K., Kajita, S., Kaneko, K., Sola, J., Yoshida, E., Mansard, N., Yokoi, K., Laumond, J.P.: Reactive stepping to prevent falling for humanoids. In: 9th IEEE-RAS International Conference on Humanoid Robots, Humanoids 2009, pp. 528–534 (December 2009)
17. Morisawa, M., Kanehiro, F., Kaneko, K., Mansard, N., Sola, J., Yoshida, E., Yokoi, K., Laumond, J.: Combining suppression of the disturbance and reactive stepping for recovering balance. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3150–3156 (October 2010)
18. 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), 677–699 (1985)
19. 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/`