Learning and Generalization of Compensative Zero-Moment Point Trajectory for Biped Walking

Kai Hu, *Student Member, IEEE,* Christian Ott, *Member, IEEE* and Dongheui Lee, *Member, IEEE*

***Abstract*—This work presents an online learning framework for improving the robustness of zero-moment point (ZMP) based biped walking controllers. The key idea is to learn a feedforward compensative ZMP (CZMP) trajectory from measured ZMP errors during repetitive walking motions by applying iterative learning control theory. The learned CZMP trajectory adjusts the reference ZMP and reduces the effect of unmodeled dynamics at the pattern generation stage. From individual learned CZMP trajectories of typical walking parameters, we can build up a CZMP database. This database can be used for generating an initial CZMP whenever a new walking pattern is executed. Prediction from the database is done by k-NN regression based on the Mahalonobis distance. Compared with state-of-the-art model-based methods, the proposed learning approach is model free and allows online adaptation to constant unknown disturbances. Enhanced walking robustness can be observed from reduced average ZMP error, and more robust reaction against external disturbances on the DLR humanoid robot TORO.**

***Index Terms*—Biped walking control, iterative learning control, balance control.**

I. Introduction

The concept of zero-moment point (ZMP) has been widely adopted as a stability indicator for gait analysis, synthesis and control of diverse anthropomorphic locomotion mechanisms since early 1970s [1]. Based on ZMP and the development of fully actuated humanoid robots many successful biped walking controllers have been demonstrated [2], [3], [4], [5]. In this control scheme a dynamically stable walking pattern is first generated and the walking control relies on precise execution of the planned trajectory. Early works plan the walking pattern offline based on accurate robot dynamics model [2], [4]. In recent years researchers have realized online walking control by utilizing simplified models for pattern generation [5], [6], [7]. The drawback of the online approaches is the inconsistency between the robot multi-body dynamics and the simplified models [8], which will cause ZMP deviation even when the planned walking pattern is executed faithfully. Combining with a feedback stabilization becomes a standard way to ease the problem of the unmodeled dynamics as well as to resist external disturbances [3], [9], [10], [11]. However they are designed to maintain balance at every control moment, rather than to solve the modeling problem of pattern generation. The dynamically inconsistent walking pattern cause persistent ZMP tracking error even when a feedback balancer is applied.

In this work we focus on the problem of improving walking robustness by refining the walking pattern, i.e. reducing the effect of unmodeled dynamics at pattern generation stage. Our key idea is to learn a feedforward compensative ZMP (CZMP) term through the measured ZMP error during the repetition of the walking trials. Because locomotion behavior displays a dominant repetitive phase, we adopt iterative learning control (ILC) theory [12], [13] as the learning strategy and name our approach as online iterative learning control of ZMP (ZMP-OILC) [14]. The learned CZMP trajectory adjusts the ZMP reference and reduces the ZMP error caused by the model inconsistency of online pattern generation. The improvement of walking robustness is evaluated qualitatively by the reduced average ZMP error and intuitively by the better resistance to external disturbances. The learned knowledge of the learning process can furthermore be used for building up a CZMP database of typical walking parameters. The CZMP trajectories of new walking motions are predicted by k-NN regression and applied as a precompensation for the online learning. Better performance such as higher convergence speed and reduced ZMP error during non-repetitive transition phases between different walking motions are achieved.

Earlier work of learning control for biped walking was presented in [15] to learn the necessary feedforward compensation in order to overcome mechanical disturbances and follow the walking pattern precisely. Later Li *et al.* proposed an algorithm to learn a trunk compensation motion for walking stabilization [16]. However the approach only works offline for a complete walking sequence. In contrast our method works online with two footsteps as one iteration, which makes the learning flexible and efficient. State-of-the-art ZMP compensation method at the pattern generation stage is the dynamical filter (ZMP-DF) [5], which computes the expected ZMP error from multi-body model and applies again the preview control for compenstation. Nishiwaki *et al.* extended the ZMP-DF by updating the walking pattern frequently from current robot status with short cycle time [17]. Compared with these methods, our approach is model free, with lower computational cost and implementation complexity. The effect of ZMP-DF is similar to conduct one learning iteration of ZMP-OILC based on the computed ZMP error from multi-body model. Moreover ZMP-OILC is able to adapt online to external constant disturbances and these disturbances are reflected to the walking pattern directly through the ZMP measurement rather than measuring the external wrench and computing the corresponding ZMP error from the dynamics model. ILC implements an integral action along the iteration domain and thus allows to improve the tracking performance at every time instant along the periodic trajectory. Notice that this is different from an integral action in time domain which allows to compensate for steady state errors but has less benefits for the transient behavior. In the context of walking control, a feedback controller with integral action has been applied e.g. in [18] in order to suppresses the steady tracking errors under presence of constant disturbances like a measurement error in the center of mass. Finally the proposed learning framework should be considered as a supplement to the feedback balancer rather than a counterpart since their combination is easy and effective.

The rest of the paper is organized as follows. In Sec. II a short review of ILC theory is given. The proposed learning framework is explained in detail in Sec. III. Learning CZMP database and its generalization are discussed in Sec.IV. Simulation and experimental results on the DLR humanoid robot TORO are shown in Sec. V to verify the effectiveness of the approach. The last section concludes the presented work.

II. Iterative Learning Control

Iterative learning control is a well studied research topic which is categorized as one branch of intelligent control [19]. The goal of ILC is to achieve better tracking performance than conventional feedback controllers through learning from previous control trials. It assumes deterministic system dynamics and repeatability of the target tracking task over a finite control horizon. Assume we have a tracking task

Kai Hu is with the Department of Electrical and Computer Engineering, Technical University of Munich (TUM), Munich 80333, Germany (e-mail: kai.hu@tum.de).

Christian Ott is with the Institute of Robotics and Mechanics, German Aerospace Center (DLR), Wessling 82234, Germany (e-mail: christian.ott@dlr.de).

Dongheui Lee is with the Department of Electrical and Computer Engineering, Technical University of Munich (TUM), Munich 80333, Germany (e-mail: dhlee@tum.de).

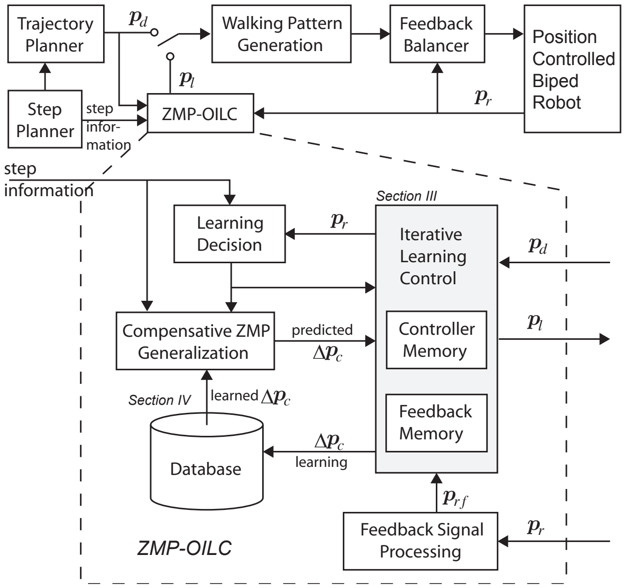

Fig. 1. Overview of online learning framework for ZMP-based biped walking controller.

with control input u and output error $e = y_d - y$, the P-type ILC update law with a forgetting factor can be written as

$$u_{i+1}(t) = u_0(t) + k_f[u_i(t) - u_0(t)] + k_l e_i(t), \quad t \in [0, T_{iter}], \tag{1}$$

in which the subscript i is the iteration number and T_{iter} is the iteration period. The learning time variable t is reset to 0 at the beginning of each iteration. $k_l > 0$ and $0 < k_f \leq 1$ are the learning gain and the forgetting factor. In the contraction mapping based method the learning convergence is predominantly determined by the static mapping $y = du$ between the system input and output. The convergence condition of (1) is given as

$$\left| k_f - k_l d \right| < 1. \tag{2}$$

A system needs to be global Lipschitz continuous to design the stable learning gain, namely the boundness of d is required. A forgetting factor smaller than one makes the learning more robust. For detailed and further information about ILC readers can refer to literature such as [13].

III. Online Iterative Learning Control of ZMP

The proposed ZMP-OILC framework can be regarded as a feedforward augment of the conventional ZMP-based online walking controllers (see the upper part of Fig. 1). As base line algorithms we adopt the preview control based pattern generator [5] and the online feedback balancer proposed by Choi *et al.* [10][1]. The details of ZMP-OILC are shown in the dashed frame in Fig 1. The rest of this section will explain each component of the online learning process.

A. Learning Decision

In this work a sequence of walking steps with same parameters is denoted as a repetitive phase. As a prerequisite for ILC our target motion should have dominant repetitive phases. For different repetitive phases walking parameters, such as step length, turning angle and step time, can vary. During the transition between different repetitive phases the motion is not repetitive. The learning controller should be aware of the step information and change its behavior at the right time. We construct the learning decision block as a finite state machine (see Fig. 2) to model different states of the learning controller and guide the learning control to start, stop and transit.

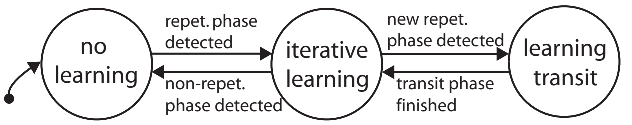

Fig. 2. Finite state machine for guiding the online iterative learning process.

B. Learning Control Law

The ZMP dynamics equation of linear inverted pendulum model (LIPM) [8] is written as:

$$\boldsymbol{p}_s = \boldsymbol{c} - \frac{c_{z,m}}{g}\ddot{\boldsymbol{c}} \tag{3}$$

in which $\boldsymbol{c}$ is the position of the COM, $c_{z,m}$ is the fixed COM height of the model and g denotes the gravity constant. $\boldsymbol{p}_s$ represents the ZMP and the subscript s stands for simplified. The measured robot ZMP $\boldsymbol{p}_r$ is considered as a sum of the part of LIPM and an unmodeled error term:

$$\boldsymbol{p}_r = \boldsymbol{p}_s + \boldsymbol{e}_{p,m} \tag{4}$$

where $\boldsymbol{e}_{p,m}$ (m stands for model) denotes the effect of model inaccuracy in the form of ZMP error. The goal of ZMP-OILC is to learn a feedforward signal $\boldsymbol{p}_l$ (l stands for learning) for compensating $\boldsymbol{e}_{p,m}$ by adjusting the reference ZMP trajectories. If $\boldsymbol{p}_l$ converges to $\boldsymbol{p}_d - \boldsymbol{e}_{p,m}$ and the pattern generator can track the reference ZMP sufficiently accurate ($\boldsymbol{p}_s \approx \boldsymbol{p}_l$ with sufficiently long preview length), then the measured robot ZMP $\boldsymbol{p}_r$ converges to the desired value $\boldsymbol{p}_d$.

In this work the learning iteration is defined as two successive footsteps[2]. The difference to the standard ILC is that the learning process is only repetitive in the local coordinate system $\boldsymbol{G}_i$ of each iteration[3]. In the global coordinate system $\boldsymbol{G}_0$ the reference ZMP trajectory $\boldsymbol{p}_{d,i}$ of each iteration varies (see Fig. 3). Consider planar walking on a flat surface with the x and y coordinates as well as the yaw turning angle α, the learning control law of ZMP-OILC based on the P-type linear ILC with a forgetting factor (1) represented in the global coordinate system $\boldsymbol{G}_0$ can be written as:

$$\begin{aligned} \boldsymbol{p}_{l,i}(t) = \boldsymbol{p}_{d,i}(t) + k_f \boldsymbol{R}_\Delta \left[\boldsymbol{p}_{l,i-1}(t) - \boldsymbol{p}_{d,i-1}(t)\right] \\ + k_l \boldsymbol{R}_\Delta \left[\boldsymbol{p}_{d,i-1}(t) - \boldsymbol{p}_{rf,i-1}(t)\right], \ t \in [0, T_{iter}], \quad i > 0 \end{aligned} \tag{5}$$

where

$$\boldsymbol{R}_\Delta = \boldsymbol{R}_z(\Delta\alpha) = \begin{bmatrix} \cos\Delta\alpha & -\sin\Delta\alpha \\ \sin\Delta\alpha & \cos\Delta\alpha \end{bmatrix}. \tag{6}$$

$\boldsymbol{p}_l$, $\boldsymbol{p}_d$ and $\boldsymbol{p}_{rf}$ represent the learned, desired as well as the filtered and timely aligned ZMP measurement (Sec. III-D) respectively. $\boldsymbol{R}_\Delta$ rotates the learning information of the last iteration to the current iteration according to walking turning angle $\Delta\alpha = \alpha_i - \alpha_{i-1}$. The second term of the right hand side represents the learned information of the previous $i-2$ iterations which is rotated by $\boldsymbol{R}_\Delta$ and weighted by the forgetting factor k_f. The third term represents the rotated ZMP error of the last iteration weighted by the learning gain k_l, which defines how much we should learn from the sensor data.

[1] A short revision of ZMP preview controller and the applied online balance controller is provided in Appendix. The proposed ZMP-OILC works also for other online pattern generation algorithms, such as [6]. It is chosen for the purpose of easy comparison with ZMP-DF.

[2] The biped system is not necessarily symmetric. This implies that the the left and right steps are taken as different even with same walking parameters.

[3] In this work the x and y axes refer to the global coordinate system $\boldsymbol{G}_0$. For local coordinate system $\boldsymbol{G}_i$ the axes are denoted as saggital (sa_i) and lateral axes (la_i).

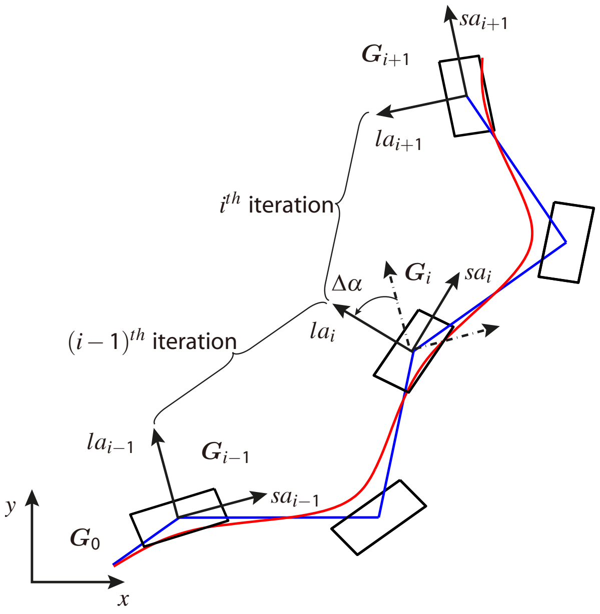

Fig. 3. Learning iteration defined for ILC.

C. Continuity of Learning Process

The learning process of ZMP-OILC is conducted continuously without reset between successive iterations. Therefore it is important to ensure the continuity. Otherwise the discontinuity will propagate throughout the whole learning process due to the integration nature of ILC. During each iteration equation (5) is a linear combination of continuous signals $\boldsymbol{p}_d$, $\boldsymbol{p}_{rf}$. The continuity of the learning process can be confirmed if the connecting points between consecutive iterations are continuous. It can be shown that if the continuity at the beginning of the first learning iteration is ensured, the whole learning process is continuous. An initialization iteration is designed to transit smoothly from non-learning to learning phases:

$$\boldsymbol{p}_{l,1}(t) = \boldsymbol{p}_{d,1}(t) + k_{l,ini}(t)\boldsymbol{R}_\Delta\left[\boldsymbol{p}_{d,0}(t) - \boldsymbol{p}_{rf,0}(t)\right] \tag{7}$$

in which $k_{l,ini}$ changes from 0 to desired value of k_l. Similarly we can exit the learning in the last iteration by designing the k_f and k_l smoothly decreasing to 0.

D. Time Alignment of Feedback Signal

ILC is essentially a point-wise integration operation of the control error along the iteration domain. Therefore the accuracy of the time alignment of the feedback signal is important. Otherwise the learning information, i.e. the control error will be computed wrongly. The real ZMP of the robot lags behind the reference due to the structural compliance and motion control [11]. An additional low pass filter is used in order to get a cleaner ZMP signal for learning. The total time lag between the filtered and the reference ZMP signal T_{delay} is identified offline by experiment. Since pattern generation needs future ZMP information, the following time condition should be fulfilled in order to achieve online consecutive learning:

$$T_{iter} \geq T_{pre} + T_{delay} \tag{8}$$

where T_{pre} represents the preview time of the pattern generator. If this time condition is not satisfied, the corresponding $\boldsymbol{p}_l(t)$ of the current filtered ZMP feedback is already in the preview buffer and cannot be modified for current iteration. In this work T_{iter} is set to be the time of two footsteps. T_{pre} is chosen as 0.1s shorter than T_{iter} since the experimentally identified T_{delay} is always within 0.1s. It is preferable to choose the preview length as long as possible under the constraint (8) in order to minimize the effect of the pattern generation error.

E. Convergence Condition

It is quite complex to analyze the convergence property with the whole system dynamics, since it contains pattern generator, motion controller and robot dynamics. One of the advantages of ILC is that it requires very little system knowledge in order to guarantee the learning convergence in the iteration domain. In our framework if we consider the system as a whole, with the desired ZMP trajectory as input and measured robot ZMP as output, the static process gain d has a value around 1, since the robot ZMP will follow the desired trajectory with some small deviations (smaller than the size of the support polygon). Therefore a selection of $0 < k_l < k_f + 1$ should lead to the convergence of (5) according to (2). A more appropriate k_l will be verified through simulation and experiment. The convergence can be evaluated by the average ZMP deviation from the desired value during one iteration:

$$\boldsymbol{e}_i = \frac{1}{T_{iter}}\int_0^{T_{iter}} \left|\boldsymbol{p}_{d,i}(t) - \boldsymbol{p}_{rf,i}(t)\right| dt \tag{9}$$

If the error variation between two iterations is sufficiently small, we conclude that the learning process converges.

IV. Generalization of Learned Compensative ZMP

It is common that humans apply knowledge gained from one task to different but related tasks. With the similar concept we first learn a database of CZMP for typical walking parameters and then generalize to different walking parameters by k-NN regression. In this way we can predict the CZMP without further learning and use the predicted trajectory as a precompensation.

A. Learning the CZMP Database

After the ZMP-OILC converges for a specific walking motion, we can extract the compensative ZMP trajectory in a learning iteration as the acquired knowledge:

$$\Delta\boldsymbol{p}_c(t) = \boldsymbol{R}_i^T\left[\boldsymbol{p}_{l,i}(t) - \boldsymbol{p}_{d,i}(t)\right] \tag{10}$$

where $\boldsymbol{R}_i^T$ represents the transpose of the rotational matrix of $\boldsymbol{G}_i$. Since $\Delta\boldsymbol{p}_c$ is a weighted time integration of sensor data, it is wise to compute the mean over several iterations after convergence to reduce the influence of sensor noise and possible non-repetitive disturbances. The database is then constructed as a lookup table of CZMP trajectories associated with typical walking parameters.

B. Generalization by k-NN Regression

Assume we have constructed the database, which consists of a sample parameter vector set $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_i | i = 1, \cdots, m\}$ and their corresponding function values $f(\boldsymbol{\theta}_i)$. The function value is parametrized by an n dimensional parameter vector $\boldsymbol{\theta}_i = \left[\theta_{1,i}, \cdots, \theta_{n,i}\right]^T$. The total number of samples m depends on parameter ranges and the necessary density of the sample parameters. Given a new target parameter vector $\boldsymbol{\theta}^\star = \left[\theta_1^\star, \cdots, \theta_n^\star\right]^T$, a k-Nearest Neighbors (k-NN) regression based on the Mahalanobis distance conducts following steps to predict the value of $f(\boldsymbol{\theta}^\star)$:

- Select k closest parameter vectors as regression candidates according to the Mahalanobis distance:

$$d(\boldsymbol{\theta}_i, \boldsymbol{\theta}^\star) = \sqrt{(\boldsymbol{\theta}_i - \boldsymbol{\theta}^\star)^T \boldsymbol{\Sigma} (\boldsymbol{\theta}_i - \boldsymbol{\theta}^\star)} \tag{11}$$

in which $\boldsymbol{\Sigma}$ is the covariance matrix of parameter vector set $\boldsymbol{\Theta}$.

Fig. 4. DLR humanoid TORO used for the experiment.

- Compute the weight of each candidate sample:

$$\omega(\boldsymbol{\theta}_i) = \frac{1/d(\boldsymbol{\theta}_i,\boldsymbol{\theta}^\star)}{\sum_{j=1}^{k} 1/d(\boldsymbol{\theta}_j,\boldsymbol{\theta}^\star)},\ i = 1,\cdots,k. \tag{12}$$

- Predict the function value as the weighted sum of the k nearest neighborhoods' function values:

$$f(\boldsymbol{\theta}^\star) = \sum_{i=1}^{k} \omega(\boldsymbol{\theta}_i) f(\boldsymbol{\theta}_i). \tag{13}$$

In this work the target value function $\boldsymbol{p}_c(t;\boldsymbol{\theta})$ is a finite trajectory. We can retrieve the predicted trajectory by time rescaling:

$$\Delta\boldsymbol{p}_c(t;\boldsymbol{\theta}^\star) = \sum_{i=1}^{k} \omega(\boldsymbol{\theta}_i)\Delta\boldsymbol{p}_c(\frac{tT_{s,i}}{T_s^\star};\boldsymbol{\theta}_i),\ t \in [0, T_s^\star]. \tag{14}$$

The difference between the right hand side of (13) and (14) is that we have to interpolate between trajectories with different time length.

C. Precompensation with CZMP

The predicted CZMP is utilized as a precompensation additional to the online learning process and achieves benefits such as faster convergence rate and reduced ZMP error during transition between different motions. The learning control law with $\Delta\boldsymbol{p}_c$ can be formulated as:

$$\begin{aligned}\boldsymbol{p}_{l,i}(t) = {} & \boldsymbol{p}_{d,i}(t) + \boldsymbol{R}_i\Delta\boldsymbol{p}_c(t) \\ & + k_f(t)\boldsymbol{R}_\Delta\left[\boldsymbol{p}_{l,i-1}(t) - \boldsymbol{p}_{d,i-1}(t) - \boldsymbol{R}_{i-1}\Delta\boldsymbol{p}_c(t)\right] \\ & + k_l(t)\boldsymbol{R}_\Delta\left[\boldsymbol{p}_{d,i-1}(t) - \boldsymbol{p}_{rf,i-1}(t)\right],\ i > 0.\end{aligned} \tag{15}$$

Compared to (5) a precompensation term $\boldsymbol{R}_i\Delta\boldsymbol{p}_c(t)$ is added.

V. Simulation and Experiment

The ZMP-OILC framework is evaluated on the DLR humanoid robot TORO (Fig. 4), which is about 1.7m tall and 75kg heavy [20]. For the walking setup we control the 12 DoFs of the legs in joint position control mode with fixed upper body joints. The ZMP is measured through the force torque sensors mounted in the feet. The proposed ZMP-OILC framework is verified both in dynamics simulation and on the real robot platform. For comparison different control strategies (listed in TABLE I) for ZMP compensation are applied. The meaning of the abbreviations in TABLE I is as follow: BL stands for base line algorithms; P stands for precompensation with CZMP; DF stands for dynamical filter with an additional preview control of 0.7s; IL stands for iterative learning.

A. Simulation

In order to verify the convergence of the ZMP-OILC and select appropriate learning parameters, we first conduct simulation with the multibody dynamics model of the TORO robot in OpenHRP3 [21].

TABLE I
CONTROL STRATEGIES OF ZMP COMPENSATION

control strategy	dynamical filter	online ILC	precompensation $\Delta\boldsymbol{p}_c$
BL	no	no	no
BLP	no	no	yes
DF	yes	no	no
IL	no	yes	no
ILP	no	yes	yes

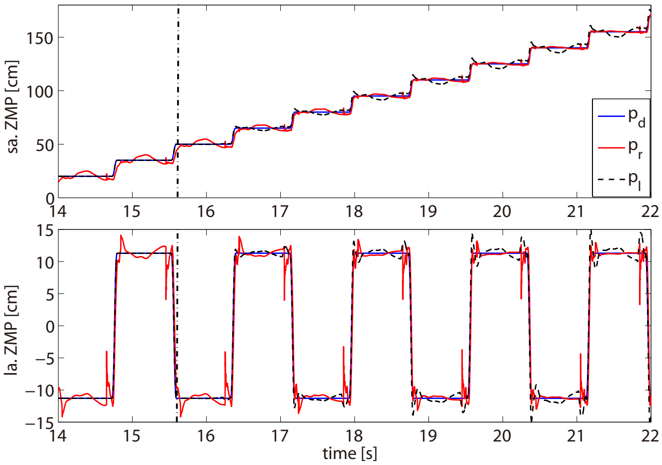

Fig. 5. Simulation result of straight forward walking with step length of 15cm, step time of 0.8s and learning parameters $k_l = 0.5, k_f = 1$. The vertical dash-dotted line marks the start time of the learning.

1) Convergence and Learning Parameters: Forward walking motion with step length of 15cm and step time of 0.8s is simulated with respect to different k_l and constant forgetting factor $k_f = 1$. Fig. 5 shows the resulted ZMP trajectories for $k_l = 0.5$. The average ZMP errors during each iteration with respect to different learning gains from 0.3 to 1.7 are shown in Fig. 6. As discussed in Sec. III-E, the learning controller should converge with $0 < k_l < 2$. With a higher value of k_l, the controller converges faster but with larger average ZMP error. In simulation the ZMP-OILC converges until $k_l = 1.3$ but starts to diverge for $k_l = \{1.5, 1.7\}$.

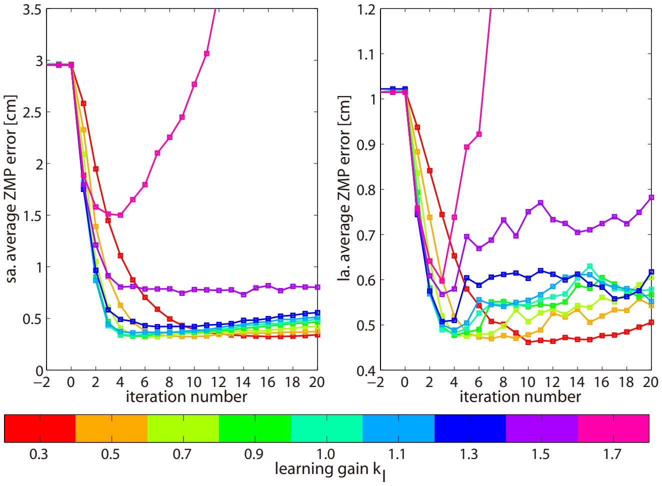

Fig. 6. Average ZMP error in each walking iteration of straight forward walking with step length of 15cm, step time of 0.8s with respect to different learning gains and constant $k_f = 1$ in simulation.

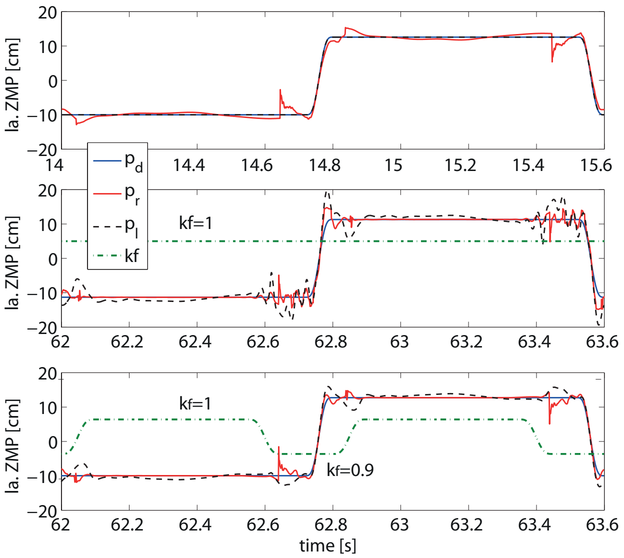

Fig. 7. Illustration of the varying forgetting factor method. The upper graph shows the discontinuities of the ZMP measurement due to food landing and lifting. The middle and lower graphs show the learning result after 40 iterations of using constant $k_f = 1$ and varying k_f from 1 to 0.9 respectively.

2) ZMP Discontinuity and Varying Forgetting Factor: The measured ZMP has discontinuities towards the landing foot and away from the lifting foot (red line in the upper graph of Fig. 7). This ZMP estimation error is hard to be compensated through the ZMP-OILC which adjusts the reference ZMP trajectories. In simulation $\boldsymbol{p}_l$ tends to grow large and the resulted $\boldsymbol{p}_r$ tends to oscillate during those time periods (black dotted line and red solid line in the middle graph of Fig. 7). We introduce a varying forgetting factor to cope with this measurement error due to the discrete nature of foot contact events. The forgetting factor k_f is set to 1 during the single supporting phase and smaller than 1 (0.9 in the simulation) during double supporting phase in which the foot landing and lifting occur. As a result the divergence of $\boldsymbol{p}_l$ and the oscillation of $\boldsymbol{p}_r$ are well suppressed, as shown in the lower graph of Fig. 7. Therefore the learning process with varying k_f will not drift in a long term perspective compared with constant $k_f = 1$, as shown in Fig. 8.

3) Comparison of Walking Robustness: We apply short and long period external forces to the robot's waist during forward walking motion to verify the improvement of walking robustness. Simulation results of three control strategies BL, DF and IL are compared. In the case of IL external forces are exerted after the learning is converged.

Short period disturbance: This scenario considers non-repetitive disturbances, e.g. unexpected collisions with the environment. Starting from 22s a force of 65N in the saggital direction is exerted at the robot waist for a period of 0.5s which corresponds to the marginal force magnitude which leads the robot to fall down without DF compensation or learning (upper graph of Fig. 9). During the time period of external disturbance (marked as gray area), ZMP is pushed to the front edge of the supporting polygon in all three cases. After the disturbance the BL algorithm already suffered from bad landing collision (high peaks after the gray area). After the landing the ZMP lies at the edge of the new support foot for a period but BL diverges[4] while the DF and IL return back to the support polygon. Compared to the case of BL where the robot tips over, DF and IL resist the disturbance successfully and only a small twist in walking direction appears (see attached video). In the case of IL the large ZMP deviation during the disturbance is also integrated by the learning and it needs several iterations to converge again. With DF the ZMP recovers almost immediately after the disturbance. This suggests that it is better to turn off the learning after the convergence if there are non-repetitive disturbances from the environment.

Long period disturbance: This scenario considers the case that the robot dynamics is changed for a longer period e.g. carrying an unknown object. In the simulation a constant force with the value of $[30, 0, -30]N$ is applied on the robot's waist to approximate the situation of carrying an object or inaccurate mass modeling. In all three cases the robot can still maintain the balance although the ZMP has a large deviation from the desired value in case of BL and DF (Fig. 9 lower graph). IL shows the best performance since it adapts to the long period disturbance and converges to the new situation. The online adaptivity of ZMP-OILC makes it applicable to situations with varying dynamics or repeatable environment changes.

4) Learning CZMP Database: To learn the CZMP database, we specified the sample walking parameter sets for saggital and lateral straight walking (SSW and LSW) as well as circle walking, which are reasonable for the robot kinematics constraints, as given in TABLE II. For each sample more than 50 iterations are learned with $k_l = 0.5$ and varying k_f with the value from 1 to 0.9. The final $\Delta\boldsymbol{p}_c$ trajectory is taken as the average of the last 10 iterations. The learned CZMP trajectories of SSW walking motion of different walking parameters are illustrated in Fig. 10.

5) Comparison of ZMP Compensation Methods: To demonstrate the proposed ZMP-OILC framework with the CZMP database, we construct a walking sequence which consists of following four different motions:

$$\begin{aligned}
&\text{motion 1: } \{d_{sa} = 12cm, d_{la} = -6cm, T_s = 0.9s\}\\
&\text{motion 2: } \{r = 0.9m, \Delta\alpha = 15^\circ, T_s = 0.9s\}\\
&\text{motion 3: } \{d_{sa} = 14cm, d_{la} = 0cm, T_s = 1s\}\\
&\text{motion 4: } \{d_{sa} = 0cm, d_{la} = 7cm, T_s = 0.9s\}\,.
\end{aligned}$$

For comparison five different control strategies (BL, BLP, DF, IL, and ILP) are simulated for this walking sequence. During the motion transitions we apply the predicted CZMP precompensation (for BLP and ILP) as a mixture of the two motions, which is designed to smoothly transit from CZMP prediction of one motion to the other.

Fig. 11 shows the CZMP generalization result of motion 2 in sagittal direction. For circle walking parametrized with 3 parameters 8 candidates (black solid lines) are selected for generalization. The blue dashed line is the $\Delta\boldsymbol{p}_c$ obtained by IL after convergence, which can be considered as optimal trajectory. The red line is the predicted CZMP from the generalization. The similarity between the predicted and the optimal trajectories reveals the performance. The k-NN regression provides a good prediction for a new motion using a-priori learned dataset, without an online learning process. With this generalization capability, the robot can estimate suboptimal CZMP trajectories for a new task reactively. The accuracy of the CZMP prediction depends both on the density of the walking parameter sets in the database and the generalization method.

[4] In Fig. 9 the ZMP measurement looks as if it exceeds the supporting polygon in two cases: 1) The collision model in OpenHRP creates high spikes when large impact occurs; 2) The supporting polygon marked with foot edges in Fig. 9 is according to the planned contact situation. In case of large disturbance the robot feet will make unexpected contacts with the ground.

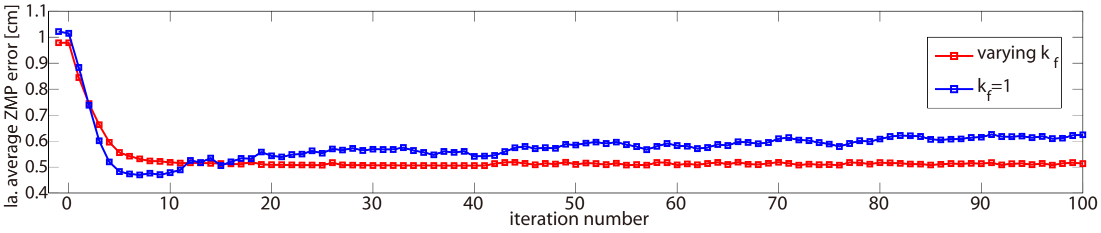

Fig. 8. Comparison of the average ZMP error with varying and constant k_f for 100 iterations of learning.

TABLE II
WALKING PARAMETER SETS FOR LEARNING THE DATABASE IN SIMULATION

Sagittal Straight Walking (SSW)	Lateral Straight Walking (LSW)	Circle Walking
$d_{sa} = \{-15,-10,-5,0,5,10,15\}\,cm$ $d_{la} = 0cm$ $T_s = \{0.6,0.8,1.0\}\,s$	$d_{sa} = 0cm$ $d_{la} = \{-7.5,-5,-2.5,0,2.5,5,7.5\}\,cm$ $T_s = \{0.6,0.8,1.0\}\,s$	$r = \{0.5,0.75,1\}\,m$ $\Delta\alpha = \{10,20\}^\circ$ $T_s = \{0.6,0.8,1.0\}\,s$

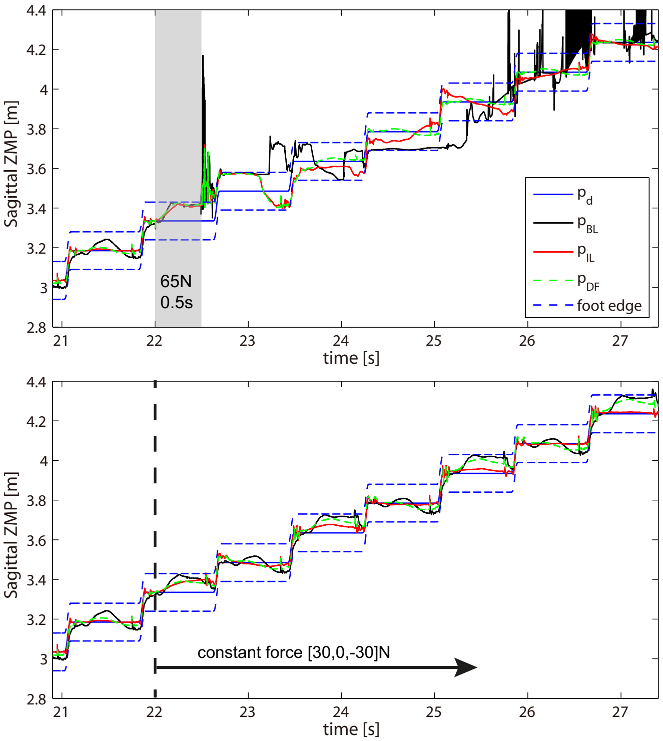

Fig. 9. Simulation result of walking robustness against short (top) and long (bottom) period external forces.

Fig. 12 shows a fraction of the walking sequence between motion 1 and 2 and the resulted ZMP trajectories in 2D view. The upper graph compares the control strategies without online learning, i.e. BL, BLP and DF. From the trajectories we can observe large ZMP errors of BL (red dotted line) and reduced ZMP errors of BLP (green solid line) and DF (black dotted line). The lower graph compares the ZMP-OILC with and without precompensation. In both cases the resulted ZMP tracks the reference well but the ILP converges faster and performs better during the transition phases. Fig. 13 shows the average ZMP error with respect to iteration number of motion 1 to 4. Based on the simulation results we can make following conclusions:

- Generalization of learned database by k-NN regression gives a good prediction for new walking motions. The performance is close to ZMP-DF.

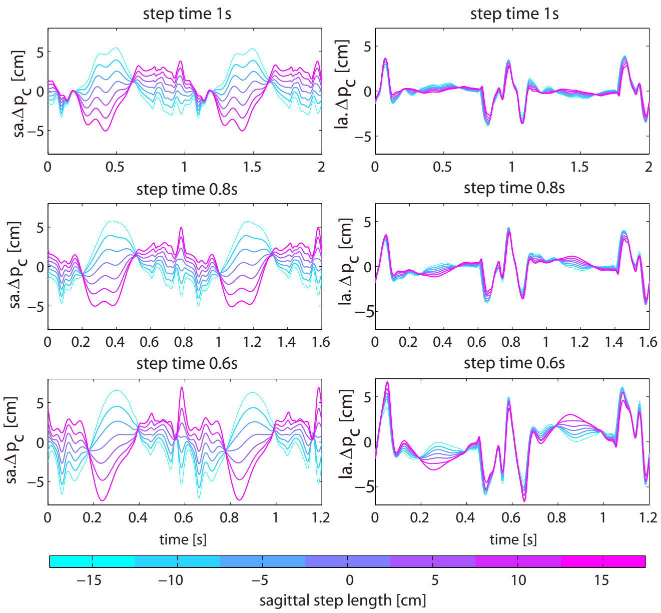

Fig. 10. Learned CZMP trajectories of sagittal straight walking motion.

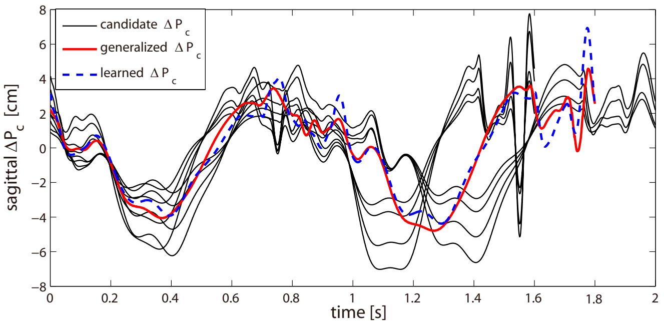

Fig. 11. Compensative ZMP trajectory prediction of motion 2 by using k-NN regression based on the Mahalanobis distance.

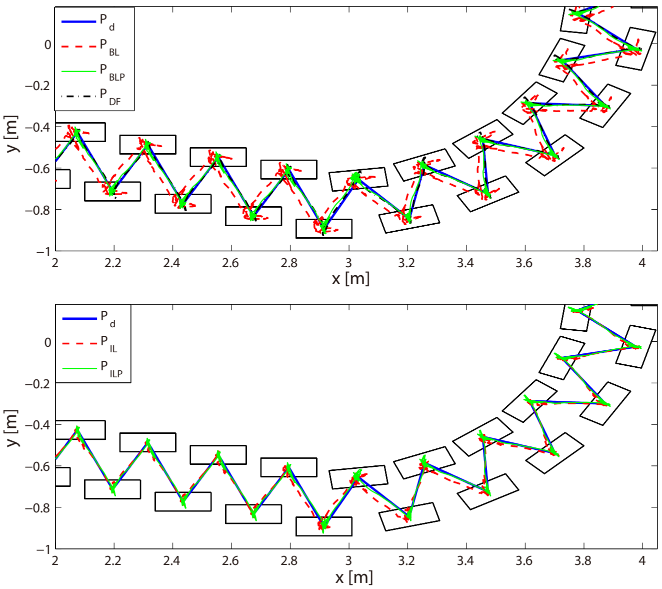

Fig. 12. A fraction of the walking sequence between motion 1 and 2 with different ZMP compensation methods in 2D view.

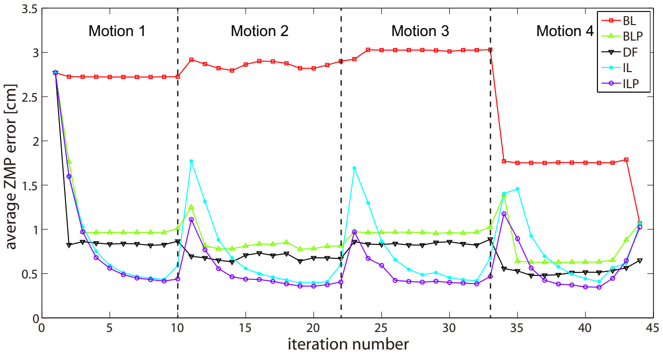

Fig. 13. The sum of average ZMP errors in both lateral and sagittal directions with respect to the learning iteration number.

- ZMP-DF can deal with non-repetitive motions such as transition phases since it is based on dynamics computation. $\Delta \boldsymbol{p}_c$ is helpful to reduce the ZMP error during the transition phases.
- Online learning (IL and ILP) results in less ZMP error after convergence compared with only applying predicted $\Delta \boldsymbol{p}_c$ or ZMP-DF.
- Applying predicted $\Delta \boldsymbol{p}_c$ or DF as the starting point of online learning improves the convergence speed.

B. Experiments

We conducted experiments of forward, backward, side walking with different step lengths on the real robot to test the ZMP-OILC algorithm. The convergence performance is influenced by the non-repetitive disturbances such as small local unevenness of the ground. Therefore smaller learning gains are applied. We set $k_{l,sa} = 0.3$ in sagittal direction and $k_{l,la} = 0.2$ in lateral direction since the distance from foot center to foot edge (safety margin of ZMP) is much smaller in the lateral direction than the sagittal direction.

The landing impact problem becomes severe in the real robot experiments since the walking control relies on high gain joint tracking control and the ground is not perfectly flat. After applying varying forgetting factor which decreases from 1 to 0.55 during double support phases the landing impact problem is solved effectively.

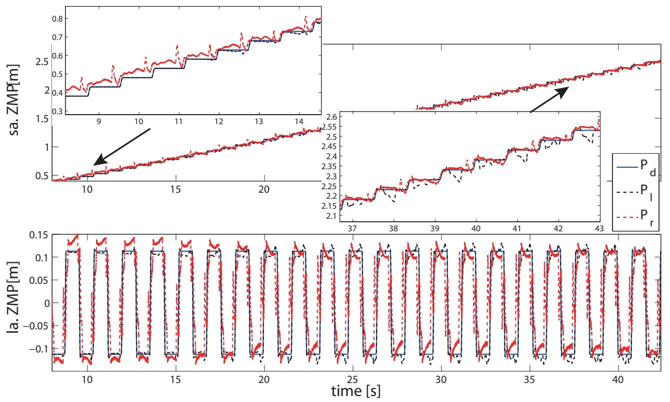

Fig. 14. Learning results on the real robot platform with $k_{l,sa} = 0.3$ and $k_{l,la} = 0.2$ and varying forgetting factor of one walking trial.

As a result of the small learning gain, the convergence rate is decreased compared with the simulation. The learning result with the varying forgetting factor of one forward walking trial is shown in Fig. 14. In real experiment the average ZMP error does not necessarily monotonically decrease in a single walking trial due to non-repetitive disturbances. In order to verify the overall convergence performance, a set of same forward walking trials are repeated. In each walking trial 20 learning iterations are performed. Figure 15 displays the mean and standard variance of the average ZMP error with respect to the learning iteration number of 10 forward walking trials. The decreasing mean value proves the effectiveness of the learning algorithm. In average the ZMP tracking error improves from 3.60cm to 1.49cm in lateral direction and from 3.31cm to 1.42cm in sagittal direction. The average standard variance is 0.44cm in lateral direction and 0.34cm in sagittal direction.

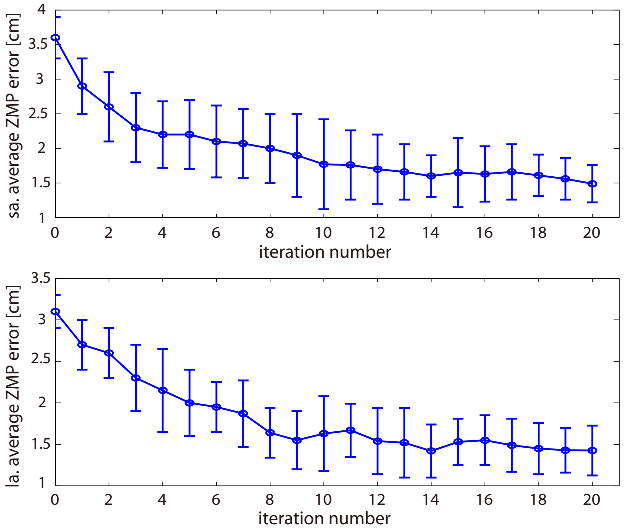

Fig. 15. Evolution of mean and standard variance of the average ZMP error with respect to iteration number of 10 forward walking trials.

VI. Conclusions

In this paper an online learning control framework for improving the robustness of biped walking is proposed. During walking repetitions CZMP trajectories can be learned online from measured ZMP error by applying iterative learning control theory, which adjusts the ZMP reference and reduces the unmodeled effects of pattern generation. Improved walking robustness is verified by lower average ZMP error and robot's better resistance to external disturbances. The online learning process can be further augmented with a CZMP database of typical walking parameters. We predict the CZMP trajectories of new walking motions by k-NN regression and apply the prediction as a precompensation. Benefits such as higher convergence speed and reduced ZMP error during non-repetitive transition phases are achieved. Compared with the classical dynamical filter approach, our method is model free, computationally light and can adapt online to unknown repetitive disturbances. As the next step we want to study about different generalization methods for CZMP prediction, e.g. extrapolation ability and application for non-repetitive walking motions.

Appendix

Pattern Generation by Preview Controller: The preview controller formulates the pattern generation as a ZMP servo problem. An optimal preview servo controller is designed based on following performance index:

$$J = \sum_{i=k}^{\infty} \left\{ Q_e (p(i) - p_{ref}(i))^2 + R(u(k) - u(k-1))^2 \right\} \qquad (16)$$

in which the weights Q_e and R are positive. $p(i) - p_{ref}(i)$ is the ZMP tracking error and $u(k) - u(k-1)$ is the incremental input. If N_l samples of future information are available, the optimal controller is given by

$$u(k) = -G_i \sum_{i=0}^{k} (p(k) - p_{ref}(k)) - G_x \boldsymbol{x}(k) - \sum_{j=1}^{N_l} G_p(j) p_{ref}(k+j), \qquad (17)$$

where G_i, G_x and $G_p(j)$ are the gains calculated from the weights Q_e, R and the system parameters of the discretized linear inverted pendulum model. In this work the preview controller is applied unmodified as in [5].

Online Balance Controller: The ZMP-COM online balancer proposed by Choi *et al.* [10] has the following feedback control law:

$$\boldsymbol{u} = \dot{\boldsymbol{c}}_d - k_p \boldsymbol{e}_p + k_c \boldsymbol{e}_c \qquad (18)$$

where $\boldsymbol{u}$ denotes the control input for the COM velocity, $\dot{\boldsymbol{c}}_d$ is the reference COM velocity vector, $\boldsymbol{e}_p$ and $\boldsymbol{e}_c$ are ZMP and CoM error vector respectively. The error gains k_p and k_c should obey the stability condition given in [10]. In our implementation the gains are set to $k_p = 2$ and $k_c = 8$.

Acknowledgment

This work is supported by Technical University of Munich - Institute for Advanced Study, funded by the German Excellence Initiative and Initiative and Networking Fund of Helmholtz Association (Grant No.VH-NG-808).

References

[1] M. Vukobratovic and B. Borovac, "Zero-moment point-thirty five years of its life," *International Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 157–173, 2004.

[2] A. Takanishi, M. Tochizawa, H. Karaki, and I. Kato, "Dynamic biped walking stabilized with optimal trunk and waist motion," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. IEEE, 1989, pp. 187–192.

[3] K. Hirai, "Current and future perspective of honda humamoid robot," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 2, 1997, pp. 500–508.

[4] S. Kagami, K. Nishiwaki, T. Kitagawa, T. Sugihara, M. Inaba, and H. Inoue, "A fast generation method of a dynamically stable humanoid robot trajectory with enhanced zmp constraint," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2000.

[5] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *IEEE Int. Conf. on Robotics and Automation*, vol. 2, 2003, pp. 1620 – 1626.

[6] S. Hong, Y. Oh, D. Kim, and B. J. You, "A walking pattern generation method with feedback and feedforward control for humanoid robots," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2009, pp. 1078–1083.

[7] P.-B. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2006, pp. 137–142.

[8] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 1, 2001, pp. 239–246.

[9] S. Kajita, K. Yokoi, M. Saigo, and K. Tanie, "Balancing a humanoid robot using backdrive concerned torque control and direct angular momentum feedback," in *IEEE Int. Conf. on Robotics and Automation*, 2001, pp. 3376–3382.

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

[11] S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro, and K. Yokoi, "Biped walking stabilization based on linear inverted pendulum tracking," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2010, pp. 4489–4496.

[12] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *Journal of Robotic systems*, vol. 1, no. 2, pp. 123–140, 1984.

[13] J.-X. Xu and Y. Tan, *Linear and nonlinear iterative learning control*. Springer Berlin, 2003, vol. 291.

[14] K. Hu, C. Ott, and D. Lee, "Online iterative learning control of zero-moment point for biped walking stabilization," in *IEEE Int. Conf. on Robotics and Automation*, 2015, pp. 5127–5133.

[15] S. Kawamura, T. Kawamura, D. Fujino, F. Miyazaki, and S. Arimoto, "Realization of biped locomotion by motion pattern learning," *Journal of Robot Society of Japan*, vol. 3, no. 3, pp. 177–180, 1985.

[16] Q. Li, A. Takanishi, and I. Kato, "Learning control for a biped walking robot with a trunk," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 3, 1993, pp. 1771–1777.

[17] K. Nishiwaki and S. Kagami, "Online walking control system for humanoids with short cycle pattern generation," *The International Journal of Robotics Research*, vol. 28, no. 6, pp. 729–742, 2009.

[18] M. Morisawa, S. Kajita, F. Kanehiro, K. Kaneko, K. Miura, and K. Yokoi, "Balance control based on capture point error compensation for biped walking on uneven terrain," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2012, pp. 734–740.

[19] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Systems*, vol. 26, no. 3, pp. 96–114, 2006.

[20] J. Englsberger, A. Werner, C. Ott, B. Henze, M. A. Roa, G. Garofalo, R. Burger, A. Beyer, O. Eiberger, K. Schmid, and A. Albu-Schffer, "Overview of the torque-controlled humanoid robot toro," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2014.

[21] F. Kanehiro, H. Hirukawa, and S. Kajita, "Openhrp: Open architecture humanoid robotics platform," *Int. Journal of Robotics Research*, vol. 23, no. 2, pp. 155–165, 2004.