Online Iterative Learning Control of Zero-Moment Point for Biped Walking Stabilization

Kai Hu, Christian Ott and Dongheui Lee

***Abstract*— Biped walking control based on simplified models relies much on online feedback stabilizers to compensate the zero-moment point (ZMP) error which partially comes from the model inconsistency of pattern generation. Inspired by the fact that human improves the performance by practicing a task for multiple times, this paper presents an online learning control framework for improving the robustness during the dominant repetitive phases of walking. The key idea is to learn a compensative feedforward ZMP term from previous ZMP error trajectories in order to achieve better ZMP tracking. Based on the iterative learning control theory, the learning process is conducted online continuously with minimal iteration of two footsteps, which can practically run in parallel with state-of-the-art walking controllers. A varying forgetting factor is designed to reduce the influence of the landing impact. Convergence of the learning control algorithm and improved ZMP tracking performance is verified both in dynamics simulation and experiment on the DLR humanoid robot TORO.**

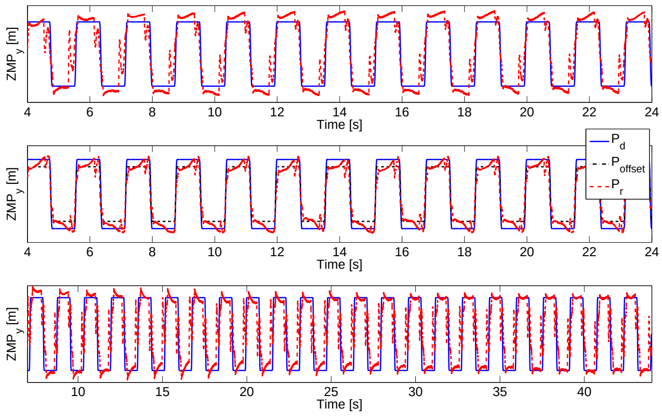

Fig. 1. Concept of learning compensative ZMP for correcting the unmodeled effect: Three graphs are recorded from real robot experiment of forward walking with 15cm step lengh. The upper graph shows a desired ZMP trajecory (blue solid line) and the resulted robot ZMP (red dotted line) with large error. In the middle graph by manully setting a constant ZMP offset (black dotted line), the ZMP deviation becomes smaller. The last graph shows the ZMP tracking improves a lot after learning from the ZMP feedback error.

I. Introduction

The concept of zero-moment point (ZMP) has been widely used for gait analysis, synthesis and control of diverse anthropomorphic locomotion mechanisms since it was proposed in early 1970s [1]. With the development of advanced humanoid robots many successful biped walking algorithms have been demonstrated by using the ZMP as the stability indicator. 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 based on accurate robot dynamics model [2], [3]. They are limited to offline applications due to the heavy computational cost. In recent years researchers have realized online walking control by utilizing simplified model such as linear inverted pendulum model (LIPM) [4] for pattern generation [5], [6]. This makes the walking control more flexible such that adaptive and reactive behaviors can be achieved [7].

The disadvantage of the online approach is the inconsistency between the real robot dynamics and the simplified models, which will cause ZMP deviation even when the robot executes the planned walking pattern faithfully. If the ZMP deviation reaches the stability margin determined by the foot geometry, the robot may start to tilt over. Therefore online feedback stabilization is designed to compensate the unmodeled effects [8], [9]. In such a successful walking case, although the real robot ZMP may not leave the support polygon, persistent ZMP tracking error exists even without external disturbances (Fig. 1 upper). In many cases hand tuning of ZMP offsets can help to achieve better stability performance (Fig. 1 middle).

There has been many research works about utilizing learning techniques to realize or improve biped locomotion. Rebula et al. presented a reinforcement learning method to obtain the Capture Point offset due to modeling errors [10]. Morimoto et al. proposed a reinforcement learning framework for adapting the foot place and walking cycle time [11]. Hu and Lee [12] proposed a method to synchronize the footsteps in walking motion imitation by learning walking primitives with Hidden Markov models. In this paper, we propose an online learning framework (ZMP-OILC) for improving the ZMP tracking performance of biped walking control. The key idea is to learn a feedforward compensative ZMP term through the actual ZMP error during the repetition of the walking trials (Fig. 1 lower). Because locomotion behavior displays a dominant repetitive phase, we apply the theory of iterative learning control (ILC), which is a well developed research topic of intelligent control [13]. It is inspired by the fact that humans learn from previous experience, e.g. athletes practice a motion multiple times in order to improve the skill. The proposed ZMP-OILC framework differs from the standard ILC that no reset is

Kai Hu and Dongheui Lee are with the Department of Electrical Engineering and Information Technology, Technical University of Munich, D-80290 Munich, Germany. kai.hu@tum.de, dhlee@tum.de
Christian Ott is with the Institute of Robotics and Mechanics, German Aerospace Center (DLR), D-82234 Weßling, Germany. christian.ott@dlr.de

required between learning cycles and the learning can be conducted online in parallel to the state-of-the-art ZMP walking controller in an efficient way with a minimal learning cycle of two footsteps.

Earlier work of ZMP learning control was presented in [14] to learn a trunk compensation motion. However the approach only works offline for a complete walking sequence. In contrast our proposed ZMP-OILC works online with learning cycle of minimal repetitive motion unit (two steps in our implementation), which makes the learning more flexible and efficient. The effect of the ZMP-OILC is similar to the ZMP dynamics filter in [5], which corrects the modeling errors through online or offline computation of multibody dynamics. The advantage of our approach is no detailed model requirement and less computational cost. Moreover since we learn from real sensor data, the result is closer to reality than relying on a multibody dynamics model. Even varying robot dynamics, e.g. robot carries an unknown object, can be handled by online learning. The learned result can be regarded as an abstraction of unmodeled system dynamics in the form of ZMP.

The rest of the paper is organized as follows. In Sec. II a short review of basics of ILC theory is given. The proposed ZMP-OILC framework is explained in detail in Sec. III. Simulation and experimental results on the DLR humanoid robot TORO are shown in Sec. IV 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 [13]. 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 with control input u and output error $e = y_d - y$, the simplest P-type ILC update law can be written as

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

where the subscript i denotes the iteration number and T_{iter} is the iteration period. k_l is the learning gain which defines the learning speed of the ILC. The learning time variable t is reset to 0 at the beginning of each iteration. One advantage of ILC is very little system knowledge requirement for designing a stable learning gain. In the contraction mapping based method the system dynamics has been frequently ignored and the learning convergence is predominantly determined by the static mapping $y = du$ between the system input and output [13]. The convergence condition in the iteration domain of (1) is given by

$$|1 - k_l d| < 1 \tag{2}$$

in which the sole required system knowledge is the equivalent process gain d. A system needs to be global Lipschitz continuous to design the stable learning gain, namely the boundness of d is required. For control task including dynamics the identical initialization condition (IIC) is required to ensure the convergence, which also implies the process repeatability. However the IIC is a critical point in practical implementations since perfect initialization is hard to achieve in real systems. As a well adopted remedy a forgetting factor is added into (1) as

$$u_{i+1}(t) = u_0(t) + k_f[u_i(t) - u_0(t)] + k_l e_i(t), \quad t \in [0, T_{iter}] \tag{3}$$

where $0 < k_f \leq 1$ is the forgetting factor. The convergence condition of (3) changes to

$$\left|k_f - k_l d\right| < 1 \tag{4}$$

However using a forgetting factor smaller than one will decrease the performance since the learned useful information is also discounted. For detailed and further information about ILC readers can refer to literature such as [13].

III. Online Iterative Learning Control of ZMP

The framework of the proposed ZMP-OILC is a feedforward augment of state-of-the-art ZMP based biped walking controller, as shown in Fig. 2. It consists of a high level learning decision block which guides the learning to initialize, start or stop, a standard ILC structure with controller memory and feedback memory, as well as an additional feedback signal processing block, which filters the noisy ZMP feedback signal and aligns the controller memory and feedback memory timely for each iteration.

A. Learning Decision

In this work a set 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. One learning iteration is defined as two successive footsteps in one repetitive phase. For different repetitive phases walking parameters can vary. Between different repetitive phases, the learning controller should be aware of the transition which is not a repetitive process. We construct a finite state machine to generate appropriate learning decision command ψ for the learning controller. The learning decision command guides the learning control to initialize, start, converge and stop.

B. Learning Control Law

The general ZMP equation considering the robot's multibody dynamics [1] can be written as:

$$\begin{aligned} p_x &= \frac{Mc_x(\ddot{c}_z + g) - M(c_z - p_z)\ddot{c}_x - \dot{L}_{c,y}}{Mc_x(\ddot{c}_z + g)} \\ p_y &= \frac{Mc_y(\ddot{c}_z + g) - M(c_z - p_z)\ddot{c}_y + \dot{L}_{c,x}}{Mc_y(\ddot{c}_z + g)} \end{aligned} \tag{5}$$

in which M is the total mass of the robot, L_c stands for the total angular momentum round the COM, p and c denote ZMP and COM quantities. By comparing the ZMP equation of the LIPM

$$\boldsymbol{p}_s = \boldsymbol{c} - \frac{c_{z,m}}{g}\ddot{\boldsymbol{c}} \tag{6}$$

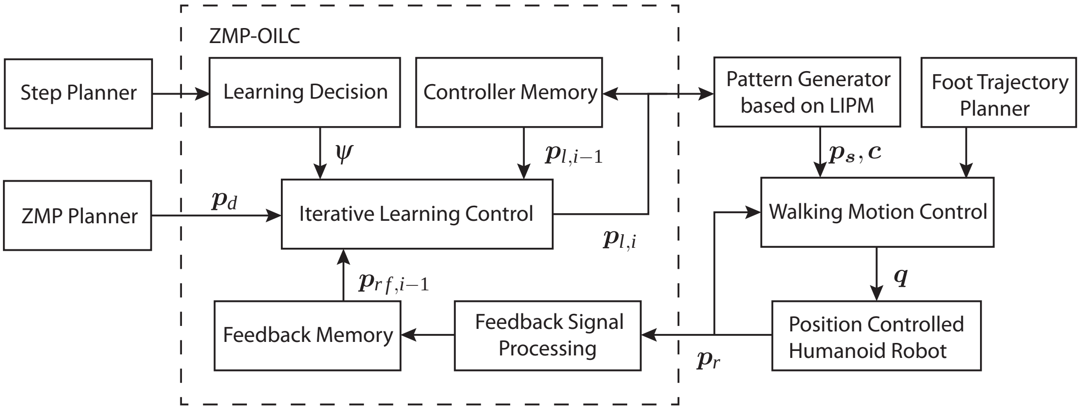

Fig. 2. Framework of online iterative learning control for biped stabilization.

in which $c_{z,m}$ is the fixed COM height of LIPM and the subscript s stands for simplified, we can represent the robot ZMP with a known term of LIPM and an unmodeled error term:

$$\boldsymbol{p} = \boldsymbol{p}_s + \boldsymbol{e}_{p,m} \tag{7}$$

where $\boldsymbol{e}_{p,m}$ (m stands for model) denotes the effect of model inaccuracy in the form of ZMP error. Dynamics filter [5] is a conventional model-based method to reduce the unmodeled effect. However it loses the simplicity of the online walking controllers which are based on simplified robot model. Despite its heavy computational cost, it is also not flexible, e.g. varying dynamics cannot be handled. In this work we tackle this problem through online learning and the above mentioned drawbacks can be resolved. Particularly we apply the ILC theory by taking advantage of the repetitive nature of walking motions.

The goal of ZMP-OILC is to learn a feedforward signal $\boldsymbol{p}_l$ (l stands for learning) in addition to the feedback control for further compensating $\boldsymbol{e}_{p,m}$. The P-type linear ILC with a forgetting factor is used as the update law in our framework:

$$\begin{aligned}\boldsymbol{p}_{l,i+1}(t) = \boldsymbol{p}_{d,i+1}(t) + k_f(\boldsymbol{p}_{l,i}(t) - \boldsymbol{p}_{d,i}(t)) \\ + k_l(\boldsymbol{p}_{d,i}(t) - \boldsymbol{p}_{rf,i}(t)),\ t \in [0, T_{iter}].\end{aligned} \tag{8}$$

$\boldsymbol{p}_{d,i}$ represents the desired ZMP for ith iteration. The value of $\boldsymbol{p}_{d,i}$ changes in each iteration according to walking distance and direction, which differs from the standard ILC. The learning process is repeatable in relative coordinates to the last iteration. The second term of the right hand side represents the learned information of the previous iterations weighted by the forgetting factor k_f. $\boldsymbol{p}_{rf}$ is the filtered and timely aligned ZMP feedback signal (Sec. III-D). The third term represents the learning of the last iteration result with the learning gain k_l, which defines how fast we should learn from the sensor data. 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.

C. Continuity of Learning Process

One difference of ZMP-OILC compared with the standard ILC is that the learning process is conducted continuously without reset between successive iterations. Therefore it is important to ensure the continuity of the learning process. Otherwise the discontinuity will propagate throughout the whole learning process due to the integration nature of ILC. During each single iteration (8) is a linear combination of continuous signals $\boldsymbol{p}_d$, $\boldsymbol{p}_{rf}$. The continuity of the learning process can therefore be confirmed if the connecting points between consecutive iterations are continuous. The end point of the ith iteration can be written as

$$\begin{aligned}\boldsymbol{p}_{l,i}(T_{iter}) = \boldsymbol{p}_{d,i}(T_{iter}) + k_f(\boldsymbol{p}_{l,i-1}(T_{iter}) - \boldsymbol{p}_{d,i-1}(T_{iter})) \\ + k_l(\boldsymbol{p}_{d,i-1}(T_{iter}) - \boldsymbol{p}_{rf,i-1}(T_{iter})).\end{aligned} \tag{9}$$

The start point of the $i+1$th iteration can be written as

$$\begin{aligned}\boldsymbol{p}_{l,i+1}(0) = \boldsymbol{p}_{d,i+1}(0) + k_f(\boldsymbol{p}_{l,i}(0) - \boldsymbol{p}_{d,i}(0)) \\ + k_l(\boldsymbol{p}_{d,i}(0) - \boldsymbol{p}_{rf,i}(0)).\end{aligned} \tag{10}$$

Since $\boldsymbol{p}_{d,i+1}(0)$ and $\boldsymbol{p}_{d,i}(T_{iter})$, as well as $\boldsymbol{p}_{rf,i+1}(0)$ and $\boldsymbol{p}_{rf,i}(T_{iter})$ are successive points along continuous signals $\boldsymbol{p}_d(t)$ and $\boldsymbol{p}_{rf}(t)$ respectively, we can conclude

$$\boldsymbol{p}_{l,i}(0) = \boldsymbol{p}_{l,i-1}(T_{iter}) \Rightarrow \boldsymbol{p}_{l,i+1}(0) = \boldsymbol{p}_{l,i}(T_{iter}). \tag{11}$$

Through back propagation, it is necessary to ensure the continuity at the beginning of the first learning iteration in order to make the whole learning process continuous. Before starting the learning process, the output of ZMP-OILC is set to $\boldsymbol{p}_{l,0} = \boldsymbol{p}_d$. When enabling the learning, the controller output of the changes from $\boldsymbol{p}_{d,0}(T_{iter})$ to

$$\begin{aligned}\boldsymbol{p}_{l,1}(0) = \boldsymbol{p}_{d,1}(0) + k_f(\boldsymbol{p}_{l,0}(0) - \boldsymbol{p}_{d,0}(0)) \\ + k_l(\boldsymbol{p}_{d,0}(0) - \boldsymbol{p}_{rf,0}(0))\end{aligned} \tag{12}$$

in which the last ZMP error term produces a discontinuity at this time instant. An initialization iteration is designed to remove this jump:

$$\begin{aligned}\boldsymbol{p}_{l,1}(t) = \boldsymbol{p}_{d,1}(t) + k_f(\boldsymbol{p}_{l,0}(t) - \boldsymbol{p}_{d,0}(t)) \\ + k_{l,init}(t)(\boldsymbol{p}_{d,0}(t) - \boldsymbol{p}_{rf,0}(t))\end{aligned} \tag{13}$$

in which the learning gain $k_{l,init}$ changes from 0 to k_l smoothly in order to achieve a smooth transition between the non-learning phase and learning phase.

D. Time Alignment of Feedback Signal

The ILC is essentially a point-wise integration operation of the control error along the iteration axis. Therefore the accuracy of the time alignment of the feedback signal is important. Otherwise the learning information, namely 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 as reported in [9]. An additional low pass filter is used in order to get a cleaner ZMP feedback signal. The total time lag can be identified offline by recording a sequence of feedback signal and computing the cross-correlation between the filtered and the reference ZMP signal.

Since most pattern generators (preview controller [5] in our implementation) need future ZMP information, the following time condition should be fulfilled in order to achieve online consecutive learning:

$$T_{iter} \geq T_{pre} + T_{delay} \tag{14}$$

where T_{pre} represents the preview time of the pattern generator and T_{delay} is the delay time of the feedback signal. If this time condition is not satisfied, the corresponding $\boldsymbol{p}_l(t)$ of the current filtered ZMP feedback is already in the preview window and cannot be modified for current iteration. This situation can be avoided by choosing a longer iteration length, e.g. integer times of two steps. It is preferable to choose the preview length as long as possible under the constraint (14) in order to minimize the effect of the pattern generation error.

E. Termination Condition of ZMP-OILC

ZMP-OILC can be always active since the computation is very simple. In practice we can also terminate the learning after it converges. The convergence can be evaluated by the average ZMP deviation from the desired value during one iteration:

$$\boldsymbol{e}_i = \frac{1}{T_{iter}} \int_0^{T_{iter}} \left| \boldsymbol{p}_{d,i}(t) - \boldsymbol{p}_{rf,i}(t) \right| dt. \tag{15}$$

If $\boldsymbol{e}_i$ is sufficiently small, we can judge that the learning process converges.

F. Iteration Domain Convergence

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 real robot ZMP as output, the process gain 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, otherwise the robot cannot walk stably). Therefore a selection of $k_f - 1 < k_l < k_f + 1$ should lead to the convergence of (8) according to (4).

Fig. 3. DLR humanoid TORO used for the experiment.

The IIC is an imperative condition for the convergence. Mismatch of the initial condition will result in tracking error and in worst case divergence of the ILC, since the initial tracking error cannot be corrected by the learning process. For the ZMP-OILC the IIC is not fulfilled since the robot state can be different at the beginning of each iteration. To ease this problem a forgetting factor is introduced into the learning control law (8) and the effect of the initial error will not accumulate.

IV. Simulation and Experimental Results

The proposed learning control framework is evaluated on the DLR humanoid robot TORO (Fig. 3). The robot is about 1.6 meters high and the body weight is about 75kg. In the experiment we control the 12 DoFs of the legs in joint position control mode with fixed upper body joints. The robot has a rather compact foot size (95mm wide and 190mm long) [15], which makes the balance control very challenging. For the ZMP measurement we utilize the force/torque sensors mounted in the robot ankles.

We test the learning framework both in dynamics simulation and on the real robot platform. A set of walking motion with different step lengths and walking directions are tested to verify the effectiveness of the ZMP-OILC. The step time is fixed as 0.8s and so the iteration time 1.6s. The preview controller proposed by Kajita et al. [5] is chosen as the pattern generator with the preview length to be 1.5s. We opt the ZMP controller in [8] as the online feedback stabilizer. A second order low pass filter with the cutting frequency of 10Hz and damping factor of 0.8 is applied to filter the ZMP measurement. The time lag of the filtered ZMP with respect to the reference trajectory is computed by applying the “finddelay” function to the recorded experiment data in Matlab. With the same low pass filter the total time lag of the filtered ZMP is 32ms in simulation and 75ms on the real robot.

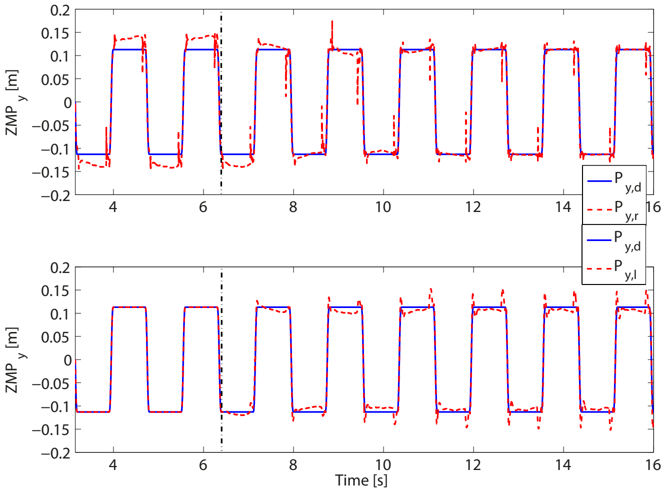

Fig. 4. Simulation result of forward walking motion with learning parameters $k_l = 0.7$, $k_f = 1$: The upper graph shows the robot ZMP (red dotted line) converges to the desired one (blue solid line). The lower graph shows the the output of ZMP-OILC (red dotted line).

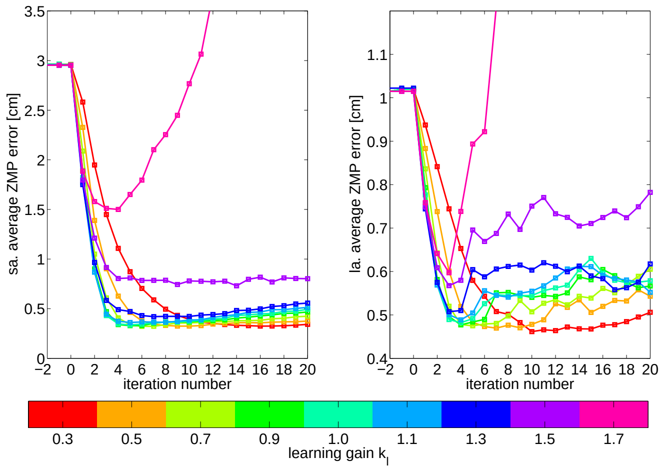

Fig. 5. Average ZMP error in each walking iteration of forward walking with 15cm step length with respect to different learning gains $k_l = \{0.3, 0.5, 0.7, 0.9, 1.0, 1.1, 1.3, 1.5, 1.7\}$ and constant $k_f = 1$ in simulation.

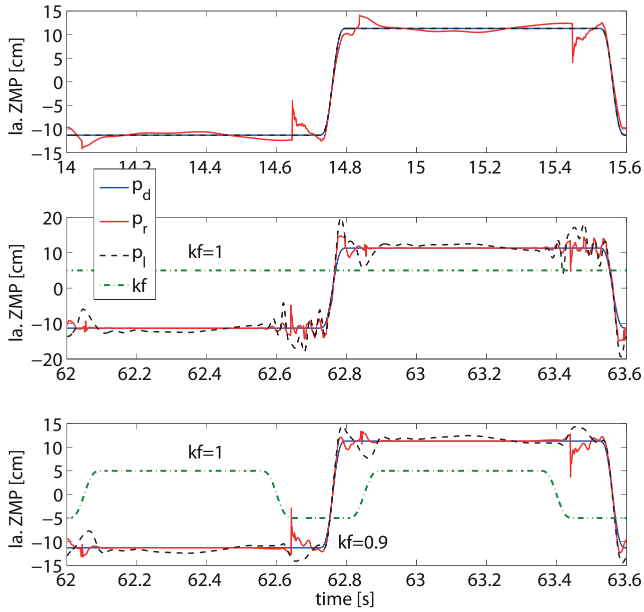

Fig. 6. Illustration of the varying forgetting factor method. The upper graph shows the discontinuities of the ZMP measurement due to food landing and lifting. The middle graph shows the learning result after 40 iterations of using constant $k_f = 1$. The lower graph shows the learning result after 40 iterations of applying varying k_f from 1 to 0.9.

A. Simulation

To verify the performance of ZMP-OILC and select appropriate learning parameters, we first conduct simulation in OpenHRP3 [16] with the multibody dynamics model of TORO robot.

1) Convergence and Learning Parameters: Forward walking of 15cm step length and 0.8s step time is simulated with respect to different learning gains. Fig. 4 shows the y directional learning result of $k_l = 0.7$. The black vertical dotted line marks the starting time of the learning control. The learning initialization takes one iteration. The upper figure shows the real ZMP tracks the desired ZMP more accurately after learning. The lower figure shows the evolution of the learned compensative ZMP trajectory for pattern generation. As shown in Fig. 5, the average ZMP tracking error (average in one learning iteration) decreases fast in the initial 5 iterations. With higher value of k_l, the controller converges faster but with larger average ZMP error. In simulation the average ZMP error in the lateral plane tends to be more unstable than in the sagittal plan. This is because the error from landing impact has larger influence. In simulation the ZMP-OILC converges until $k_l = 1.3$ but starts to diverge for $k_l = \{1.5, 1.7\}$.

2) Landing Impact and Varying Forgetting Factor: The ZMP error caused by the foot landing and lifting brings oscillations to the ZMP-OILC after long time of learning because of its integration nature. In the walking case the measured robot ZMP has jumps towards the new landing foot and away from the lifting foot (red line in the upper graph of Fig. 6). This type of error is due to the discrete nature of foot contact events and is hard be compensated through the ZMP-OILC which only adjusts the reference ZMP trajectory. However the integration of the landing and lifting ZMP error over walking iterations tends to diverge since it has same sign at the same time instant in each iteration. In simulation $\boldsymbol{p}_l$ tends to diverge and resulted ZMP $\boldsymbol{p}_r$ tends to oscillate during those time periods (black dotted line and red solid line in the middle graph of Fig. 6). Same strategy as the remedy of IIC, we introduced a varying forgetting factor to ease the influence of the ZMP error during the stance changes. 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. Smooth transitions are designed before the landing impact and after the foot lifting. 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. 6. Fig. 7 demonstrates the effectiveness of the varying forgetting factor method through long period

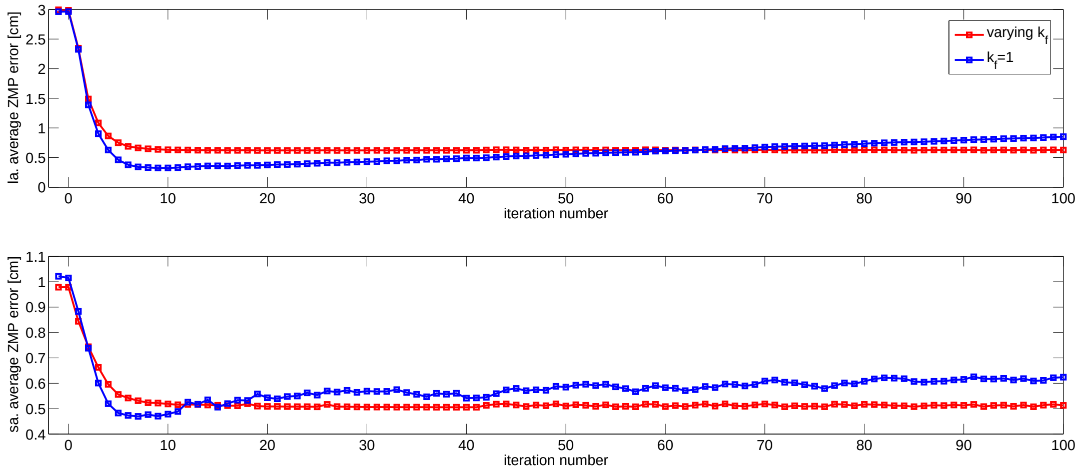

Fig. 7. Comparison of the average ZMP error with varying k_f from 1 to 0.9 and constant $k_f = 1$ for 100 iterations of learning.

of learning (100 iterations). As expected the average ZMP error of constant $k_f = 1$ reaches a smaller value in the beginning than that of varying k_f, since $k_f < 1$ will decrease the learning performance. But the error grows and tends to diverge in the long term perspective (also can be observed in Fig. 5) while with the varying k_f the error is kept almost constant.

B. Experiment

After the convergence is verified in simulation, we conduct experiments on the real robot platform. Forward, backward, side walking with different step lengths are tested. The convergence performance is influenced by the non-repetitive disturbances such as small local unevenness of the ground. Therefore smaller learning gains are applied to prevent possible divergence of the learning. We set $k_{l,x} = 0.3$ in x direction and $k_{l,y} = 0.2$ in y direction since the distance from foot center to foot edge (stable margin of ZMP) is much smaller in y direction than x 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. The structural deflection during the single support phase makes the foot landing hard to regulate by joint position control. Accumulation of this large ZMP error in the learning controller leads to divergence (Fig. 8) and the robot tilts over. In order to solve this problem we introduce a varying forgetting factor. Varying forgetting factor which decreases from 1 to 0.55 during double support phases is used. It resolves the landing impact problem effectively.

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. 9. The average ZMP error with respect to the learning iterations is shown in Fig. 10. In real experiments non-repetitive disturbances will appear and therefore the average ZMP error does not necessarily monotonically decrease in a single walking trial. In order to verify the overall convergence performance, a set of same

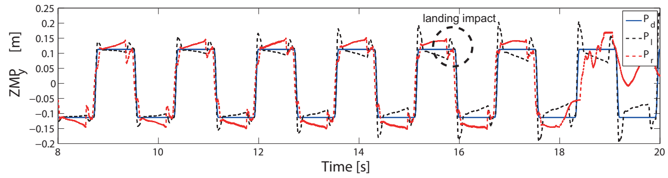

Fig. 8. Learning diverges with a constant $k_f = 1$ on the real robot because of accumulation of the landing impact ZMP error.

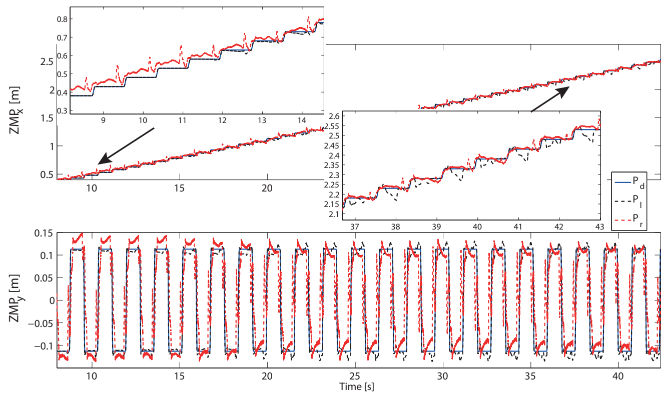

Fig. 9. Learning results on the real robot platform with the varying forgetting factor of one walking trial. The convergence rate is relative slow with small learning gain $k_{l,x} = 0.3$ and $k_{l,y} = 0.2$.

forward walking trials are repeated. In each walking trial 20 learning iterations are performed. Fig. 11 displays the evolution of the mean and standard variance of the average ZMP error with respect to the learning iteration number of 10 forward walking trials of 15cm step length. 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 x direction and from 3.31cm to 1.42cm in y direction. The average standard variance is 0.44cm in x direction and 0.34cm in y direction.

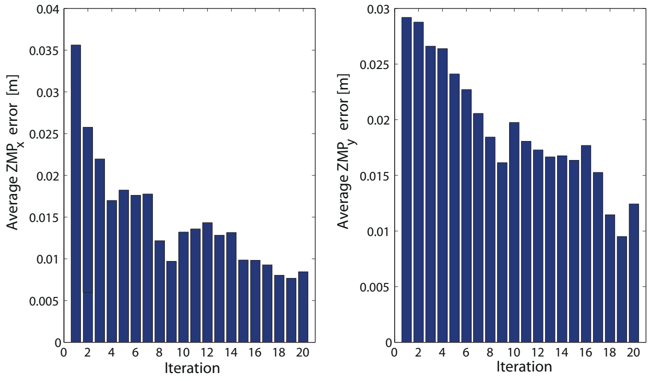

Fig. 10. Average ZMP error (average in one learning iteration) with respect to iteration number of one forward walking trial with 15cm step length.

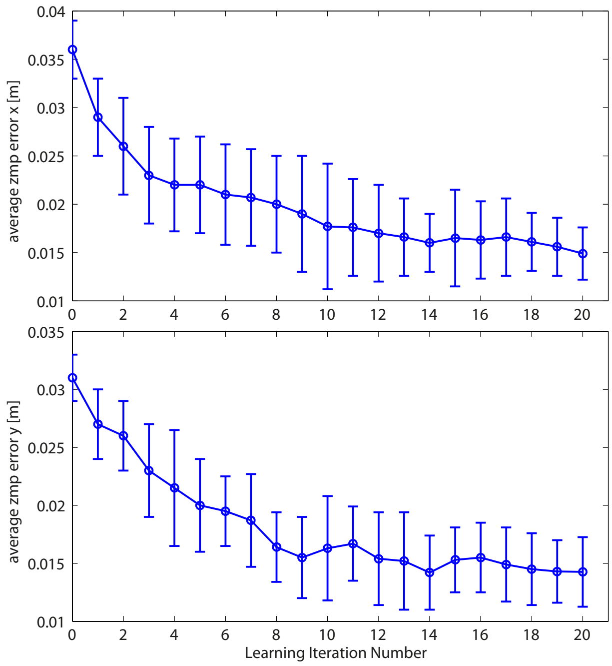

Fig. 11. Mean and standard variance of the average ZMP error with respect to iteration number of 10 forward walking trials with 15cm step length.

V. Conclusions

In this paper we propose an online learning control framework of ZMP (ZMP-OILC) for improving the robustness of biped walking. The focus is to achieve better ZMP tracking performance during the repetitive phase of walking through learning a compensative ZMP trajectory from previous ZMP tracking error. We apply a P-type iterative learning control law with forgetting factor, which is effective, computationally cheap and robust against varying initial conditions. Varying forgetting factor is designed to ease the problem of the landing impact. The proposed method can be used efficiently as a feedforward augment in addition to state-of-the-art ZMP based walking controller in an online fashion. Simulation and experiment results on the DLR humanoid robot TORO show the effectiveness of the ZMP-OILC. For future works we want to investigate the possibility of utilizing the previous learned compensative ZMP information to achieve also fast convergence on the real robot.

Acknowledgments

This work is supported partially by Technical University Munich - Institute for Advanced Study, funded by the German Excellence Initiative and also by the Initiative and Networking Fund of Helmholtz Association (Grant no. VH-NG-808). The authors want to thank Mr. Alex Werner, Mr. Bernd Henze and Mr. Johannes Englsberger for their help on conducting the experiment.

References

[1] M. Vukobratovic and B. Borovac. Zero-moment point-thirty five years of its life. *International Journal of Humanoid Robotics*, 1(1):157–173, 2004.

[2] Atsuo Takanishi, Mamoru Tochizawa, Hideyuki Karaki, and Ichiro Kato. Dynamic biped walking stabilized with optimal trunk and waist motion. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 187–192. IEEE, 1989.

[3] Satoshi Kagami, Koichi Nishiwaki, Tomonobu Kitagawa, Tomomichi Sugihara, Masayuki Inaba, and Hirochika 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.

[4] 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*, volume 1, pages 239–246, 2001.

[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*, volume 2, pages 1620 – 1626 vol.2, sept. 2003.

[6] 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*, pages 137–142, 2006.

[7] Mitsuharu Morisawa, Kensuke Harada, Shuuji Kajita, Kenji Kaneko, Joan Sola, Eiichi Yoshida, Nicolas Mansard, Kazuhito Yokoi, and J-P Laumond. Reactive stepping to prevent falling for humanoids. In *IEEE-RAS Int. Conf. on Humanoid Robots*, pages 528–534. IEEE, 2009.

[8] Y. Choi, D. Kim, Y. Oh, and B.J. You. Posture/walking control for humanoid robot based on kinematic resolution of com jacobian with embedded motion. *IEEE Trans. on Robotics*, 23(6):1285–1293, 2007.

[9] Shuuji Kajita, Mitsuharu Morisawa, Kanako Miura, Shinichiro Nakaoka, Kensuke Harada, Kenji Kaneko, Fumio Kanehiro, and Kazuhito Yokoi. Biped walking stabilization based on linear inverted pendulum tracking. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 4489–4496. IEEE, 2010.

[10] John Rebula, Fabian Canas, Jerry Pratt, and Ambarish Goswami. Learning capture points for humanoid push recovery. In *IEEE-RAS Int. Conf. on Humanoid Robots*, pages 65–72. IEEE, 2007.

[11] Jun Morimoto, Gordon Cheng, Christopher G Atkeson, and Garth Zeglin. A simple reinforcement learning algorithm for biped walking. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 3, pages 3030–3035. IEEE, 2004.

[12] Kai Hu and Dongheui Lee. Prediction-based synchronized human motion imitation by a humanoid robot. *at-Automatisierungstechnik*, 60(11):705–714, 2012.

[13] Jian-Xin Xu and Ying Tan. *Linear and nonlinear iterative learning control*, volume 291. Springer Berlin, 2003.

[14] Qinghua Li, Atsuo Takanishi, and Ichiro Kato. Learning control for a biped walking robot with a trunk. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 3, pages 1771–1777. IEEE, 1993.

[15] Christian Ott, C Baumgartner, Johannes Mayr, Matthias Fuchs, Robert Burger, Dongheui Lee, Oliver Eiberger, A Albu-Schaffer, Markus Grebenstein, and Gerd Hirzinger. Development of a biped robot with torque controlled joints. In *IEEE-RAS Int. Conf. on Humanoid Robots*, pages 167–173. IEEE, 2010.

[16] Fumio Kanehiro, Hirohisa Hirukawa, and Shuuji Kajita. Openhrp: Open architecture humanoid robotics platform. *Int. Journal of Robotics Research*, 23(2):155–165, 2004.