Proceedings of the 2017 IEEE
International Conference on Robotics and Biomimetics
December 5-8, 2017, Macau SAR, China

A Robust Biped Gait Controller Using Step Timing Optimization with Fixed Footprint Constraints

Haitao Wang
Department of Automation
Tsinghua University
Beijing, China
wht17@mails.tsinghua.edu.cn

Mingguo Zhao
Department of Automation
Tsinghua University
Beijing, China
mgzhao@mail.tsinghua.edu.cn

***Abstract*—Humans are able to wrestle each other on plum blossom pole (a kind of Chinese Kong Fu on tiny footholds), while our robots have yet to recover from a push while walking with fixed footprint constraints. We find it remarkable that combining step-timing optimization with classical modification of Center of Pressure (CoP) reduces the margin of instability. In this paper, we propose a simplified multi-objective optimization based algorithm to keep the Divergent Component of Motion (DCM) of the Linear Inverted Pendulum (LIP) model we use within stable margin, and make the robot more agile when recovering from tilted state. We have verified our algorithm robustness via dynamics simulation and hardware experimental results on our THU-Strider Platform.**

Keywords*—*Divergent Component of Moiton; Linear Inverted Pendulum;

I. INTRODUCTION

In order to make the robot achieve a stable and robust walking like human doing, we generally use the concept of Zero Moment Point (ZMP) [1] to measure whether the robot is in balance, and WABOT-1 firstly realized a dynamic walking based on ZMP [2]. However, optimizing the ZMP of the multi-body model with complex dynamics takes up numerous computing resources, which limits its performance in online walking control system. The Linear Inverted Pendulum (LIP) is a model with linear physics equation to simplify the gait generator [3]. Kajita proposed the widely used method that using preview controller with LIP and Cart-Table model to make the ZMP trajectory as predefined [4].

Furthermore, we can split the LIPM based robot's motion into stable and unstable components. J. Pratt [5] named the boundary line between them is named Instantaneous Capture Point (ICP), we only need to control the unstable part which is known as DCM to keep the balance of the robot [6]. A similar concept called Extrapolated Center of Mass (XCoM) is firstly used by Hof in robot walking gait generator [7], but DLR's Englsberger do have a great contribution to the stability analysis of DCM based walking control system[8, 9].

For the LIP model have only one mass without inertia, people have to use a modified LIP with Flywheel [10] when recovering from a large push. Since the ZMP and CoP is coincident, we use additional ankle torque which influence the Ground Contact Force to control the CoP within the support polygon of the robot, which play the same role as upper body pitching or arms waving based ZMP controller.

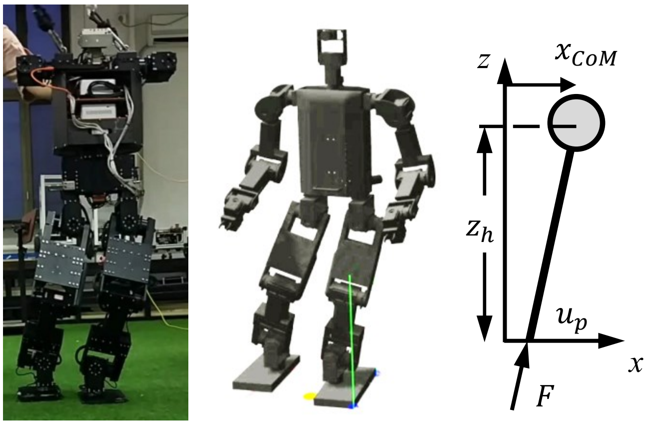

Fig. 1. Left Image shows hardware experiment in which our robot THU-Strider is being pushed forward. The right picture shows the simulation model of our platform in gazebo.

The stability margin of the DCM and ankle torque CoP controller decreases as the robot walks faster and becomes more dynamic. Griffin and other people tries step adjustment to improve robustness when facing large external force [11]. Kryczka proposed that step location and timing could be optimized together [12], which inspired by the motion of human when recovering from a large push. Khadiv approximates the nonlinear step-timing optimization problems as a linear one, which can be powerful in online controller [13].

However, there are some cases such as walking on the plum blossom pole or scattered bricks that limited foothold can be selected, which means step location is predefined and cannot be changeable. Englsberger's CP controller can only recover from small errors of CoM trajectory with fixed footprint constraints. Therefore, we propose a method to optimize the step timing and CoP together in the process of recovering. Instead of adjusting the position of the next step or current CoP, we simply set the swing foot down much more quickly or slowly before the LIP's divergent component become uncontrollable when we push or pull the robot. And we also simplify the multi-objective optimization, and found its approximate solution which can be run in real time on our Intel-NUC with an i5-5700 CPU taking no longer than 10μs.

II. Step Timing And CoP Optimization

A. Brief Introduction to LIP and DCM dynamics

We assume the LIP's height constant, with massless telescopes as legs in Fig.1.c. Its dynamics is decoupled in x/y-axis and the equation of x dimension can be formulated as (1).

$$\ddot{x} = \omega_0^2(x - u) \tag{1}$$

The CoP (u) will push CoM (x) to diverge if not changed, and the ω_0 is called as the natural frequency of the LIP ($\omega_0 = \sqrt{g/z_0}$, where g = 9.80 m/s^2, z_0 is the constant height).

$$x(t) = C_1 e^{\omega_0 t} + C_2 e^{-\omega_0 t} \tag{2}$$

We can split the dynamics of the LIP into a divergent part and a converge part in (2), the coefficients of (2) can be calculated if initial state of LIP is given, and we specify the property to find the divergent part in (3), adding a new variable ξ contains both position and velocity to link the two parts up.

$$\begin{gathered} \dot{x} = \omega_0(\xi - x) \\ \dot{\xi} = \omega_0(\xi - u) \\ \xi = x + \frac{1}{\omega_0}\dot{x} \end{gathered} \tag{3}$$

The stable part in which the CoM converges to the DCM with a stable first order dynamics is the first equation of (3), and the second of (3) shows that DCM is pushed away by the CoP with an unstable first order dynamics.

The second equation of (3) can be solved into the time domain as (4) to calculate the DCM value in the future, given the initial value of DCM and assume the CoP will keep as $u(t)$ during t_0 to T.

$$\xi(T) = \big(\xi(t_0) - u(t)\big)e^{\omega_0(T-t_0)} + u(t) \tag{4}$$

Therefore, we can control the DCM using the modifications of the CoP [14], which is proposed by Englsberger as a CP controller.

B. Step Timing Optimization

In the traditional step timing fixed DCM controller, there is a coefficient $k = \frac{e^{\omega\Delta T}-1}{e^{\omega\Delta T}}$ in the expression of $u(t)$ as (5).

$$u(t) = \frac{1}{k}\xi(t_0) + \left(1 - \frac{1}{k}\right)\xi(t_0 + \Delta T) \tag{5}$$

When the robot is being pushed by an external force, the $\xi(t_0)$ (Point C in Fig. 3) we measured will affect the $u(t_0)$ (Point E in Fig.3) in turn. The boundary of the support polygon will limit the maximum force which robot can take and do not control the CP to the desired location. The limitation is obviously in Fig.2 where the desired CoP is discontinious.

For we have fixed footprint constraints in this paper, we focus on the controllable and optimizable variables such as step timing and the CoP location. We can firstly write our linear optimization as (6)

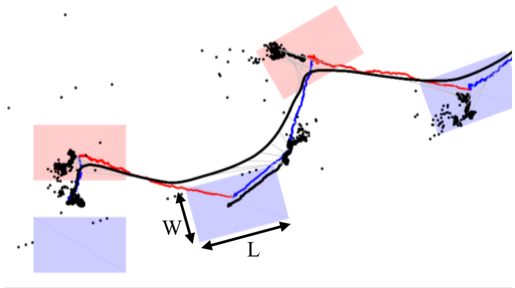

Fig. 2. The traditional DCM controller which is affected by both of the noise of the CoM estimator and the external force. W&L means width and length of the footprint constraints.

$$\begin{gathered} \min_{u_t, T, \xi_T} F = \|u_t - u_0\|_2 \\ \text{s.t.} \begin{bmatrix} 1 & & \\ -1 & & \\ & 1 & \\ & -1 & \\ & & 1 \\ & & -1 \end{bmatrix} \begin{bmatrix} u_{t,x} \\ u_{t,y} \\ e^{\omega T} \end{bmatrix} \le \begin{bmatrix} u_{0,x} + \frac{L}{2} \\ -u_{0,x} + \frac{L}{2} \\ u_{0,y} + \frac{W}{2} \\ -u_{0,y} + \frac{W}{2} \\ e^{\omega_0 T_{max}} \\ e^{\omega_0 T_{min}} \end{bmatrix} \\ \xi_{T,x} - u_{t,x} = \big(\xi_{t,x} - u_{t,x}\big)e^{\omega(T-t)} \\ \xi_{T,y} - u_{t,y} = \big(\xi_{t,y} - u_{t,y}\big)e^{\omega(T-t)} \end{gathered} \tag{6}$$

The value function can be written as a Quadratic Program (QP) to minimize the modification of the CoP from its desired trajectory. And it is found that the errors of DCM from external force parallel to the forward or backward direction can be solved using step timing optimization to find a proper CoP, but the vertical force cannot be removed by adjusting step-timing.

In Fig.3, we can see that the feasible domain of the CoP consists of a line segment, which is parallel to the line CD. It can explain that optimizing the step timing cannot remove all the effect of disturbance vertical to our moving direction, while the feasible domain need to be larger to include the original desired CoP location.

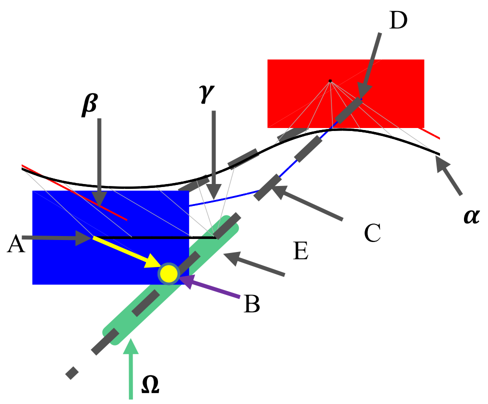

Fig. 3. A: CoP before push, $u(t_0)$; B: Optimized CoP input, u_{t_1}; C: DCM after push, ξ_{t_1}; D: the desired CP(DCM turning point), ξ_T; E: the modified CoP after push using step timing fixed DCM controller; α: CoM trajectory; β: desired DCM trajectory; γ: real DCM trajectory; Ω: the feasible domain of the CoP.

C. Multi-objective optimization Based Gait Controller

In order to expand the feasible domain of the problem, we enlarge the tolerable range of the target DCM error, and set it as a part in the value function. We can solve the problems as a multi-objective optimization equation in (7).

$$\min_{u_t,T,\xi_T} F = \alpha_1\|u_t - u_0\|_2 + \alpha_2\|\xi_T - \xi_T^*\|_2$$

$$s.t.\begin{bmatrix} 1 & & & & \\ -1 & & & & \\ & 1 & & & \\ & -1 & & & \\ & & 1 & & \\ & & -1 & & \\ & & & 1 & \\ & & & -1 & \\ & & & & 1 \\ & & & & -1 \end{bmatrix}\begin{bmatrix} u_{t,x} \\ u_{t,y} \\ e^{\omega T} \\ \xi_{T,x} \\ \xi_{T,y} \end{bmatrix} \le \begin{bmatrix} u_{0,x} + \frac{L}{2} \\ -u_{0,x} + \frac{L}{2} \\ u_{0,y} + \frac{W}{2} \\ -u_{0,y} + \frac{W}{2} \\ e^{\omega_0 T_{max}} \\ e^{\omega_0 T_{min}} \\ \xi_{T,x}^* + \Delta x \\ -\xi_{T,x}^* + \Delta x \\ \xi_{T,y}^* + \Delta y \\ -\xi_{T,y}^* + \Delta y \end{bmatrix} \quad (7)$$

$$\xi_{T,x} - u_{t,x} = (\xi_{t,x} - u_{t,x})e^{\omega(T-t)}$$

$$\xi_{T,y} - u_{t,y} = (\xi_{t,y} - u_{t,y})e^{\omega(T-t)}$$

Compared to the linear optimization we solved, there is another constraint for the target DCM location, we set it lies near the desired DCM goal, and tries to find a balance between the DCM error and the CoP modification. Fig. 4 reveals the principle of expanding the CoP feasible domain, and the Point B is much closer to original desired CoP location.

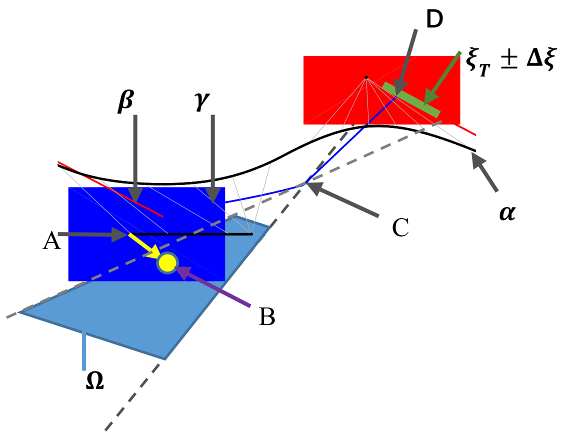

Fig. 4. A: CoP before push, $u(t_0)$; B: Optimized CoP input, u_{t_1}; C: DCM after push, ξ_{t_1}; D: the desired CP(DCM turning point), ξ_T; α: CoM trajectory; β: desired DCM trajectory; γ: real DCM trajectory; Ω: the feasible domain of the CoP(Polygon); $\xi_T \pm \Delta\xi$: the tolerable range of DCM with error.

The outputs of the multi-objective optimization are the step timing to correct and the desired CoP to control. We solved the numerical point that is closer to original CoP than before, but the time we take is longer than 0.5ms, which may affect the performance in real-time program.

D. Simplified Optimazation Problem in MATLAB Simulation

We simplify the optimization to an analytical problem that has only one variable k, for the DCM after push is on a line segment with a slope of k between point $u(t)$ and point ξ_T, and the value function we minimize is a QP. It can be proved that if $F(k)$ in (8)has a minimum value, then $G(k) - (k^2 + 1)F_{min}(k) = 0$ has double root, F_{min} is solvable by calculate the equation with the root of the discriminant equals to zero.

$$F(k) = d_{u_0\perp l}^2 + d_{\xi_T\perp l}^2 = \frac{G(k)}{k^2+1}, G(k) = \sum\|k\Delta x + \Delta y\|^2 \quad (7)$$

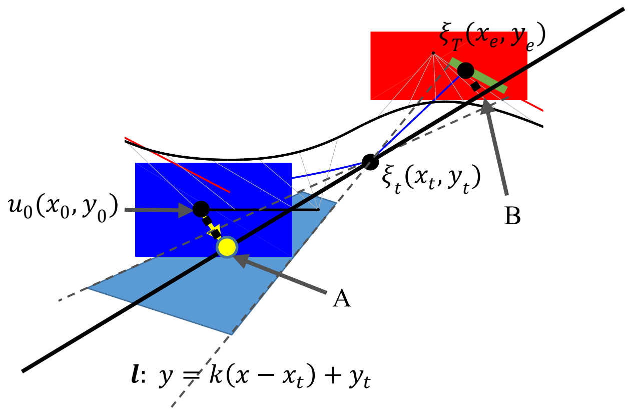

Fig. 5. The simplified solution to muti-objective optimization. A: the modified CoP which is the foot of the perpendicular from u_0 on the line l; B: the modified DCM target which is the foot of the perpendicular from ξ_T on the line l.

And the minimum solution we find using this method is actually same in most situations we probably meet, while it takes no longer than 10 μs.

Below is the result of numerical simulation of LIP when facing large push after optimizing in Fig. 6. The simulation is based on Simulink, where the closed-loop of the DCM controller can be easily built and the dynamics of LIP is simulated using numerical integratio. It can be seen that through step timing and CoP optimization, LIP model can adapt to huge external forces with fixed footholds while walking.

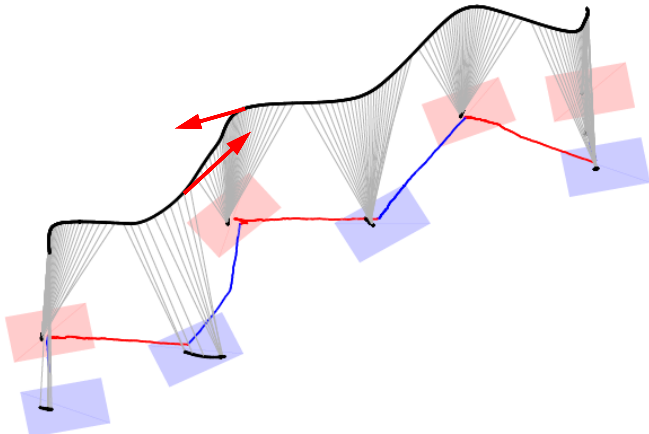

Fig. 6. The simulation of the LIP model when external force exists. Red lines and blue lines are the DCM on the ground, rectangles in bule and red are the fixed footprint which robot must step on. Red arrows are the external force to the CoM of the LIP. Black curves on the ground are the optimized CoP trajectories. Black curve above the ground is the CoM trajectory.

III. Whole Body Controll System

A feedforward and feedback control system is designed to generate a robust walking gait in Fig. 7. The DCM trajectory generator is described in section II and updated every 100ms, using a simplified optimization algorithm to calculate the time left in current step and the place where CoP is to shift to. The inner loops, including CoP loop and joint loop, tries to control the multi-body model of the robot as outer loop desire. CoP loop uses the current estimated CoP position and the CoM states of the multi-body model to calculate a compensated acceleration for the CoM controller. The CoM controller serves as a numerical integrator to send CoM states to Dynamics FF module.

$$F = M_x(\theta)\ddot{x} + V_x(\theta, \dot{\theta}) + G_x(\theta)$$
$$\tau = J^T(\theta)F \tag{8}$$

The Dynamics FF module contains calculation of stance leg with upper body's inverse dynamics and swing leg's impact on the CoM according to (8), while the Dynamics Feedback will control the errors between the multibody model and the real robot. The actuator we use is ROBOTIS DXL pro, a current controllable motor, therefore we can set a torque feedforward in the joint loop to control the CoM's acceleration as we expected.

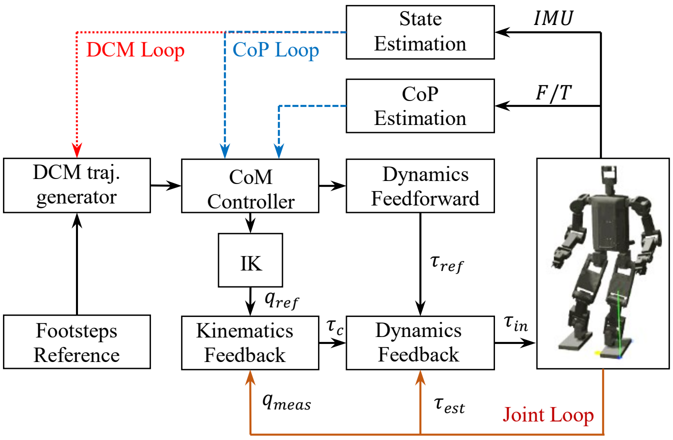

Fig. 7. Overview of the whole body control system. The DCM loop updates the DCM trajectory evrey 100ms, with the CoP loop and the inner joint hybrid dynamics controller works at a frequency of 200Hz.

A. Swing Foot Planning

The swing foot trajectory will be regenerated after the step timing updated, we set a boundary time window in the single support phase where the step timing may be modified, and we use spline interpolation to fit 9 special turning point during swing in the Cartesian space.

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{9}$$

Then we use another third order polynomials to obtain a smoother swing leg trajectory in time domain. We can avoid discontinuous changes of acceleration. The position of the swing foot can be formulated like (9) when step timing changed.

The coefficients $a_0 \dots a_3$ are determined by velocity and acceleration of foot at different period that has different time scales.

B. Dynamics Compensation Based CoP Controller

To control the CoP as we want, we use the F/T sensors on the ankle to measure the real CoP as a feedback, then we set an additional torque to the actuators of the ankle. It can be seen from the experiment that CoP controller performs well in Fig.8 compared to the same system without CoP controller.

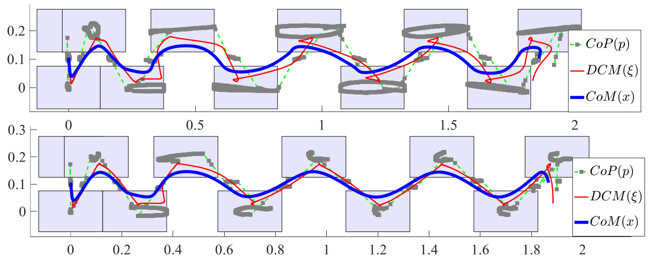

Fig. 8. Result of walking gait controllers of robot pushed forward in the third step, where we can find widings of the DCM trajectory in both figures. The upper image is the result without CoP controller, while the figure below records the stable convergence process of CoP.

IV. Experiments Results

The THU-Strider Platform we use contains a 130cm, 45kg humanoid robot Strider and a simulation platform. Strider is a fully autonomous robot equipped with Stereo Vision Module and high performance actuators with 20 DOF. The whole motion control systems runs at an intel NUC with a 2 core 2.5GHz CPU, sending the torque instruments to actuators at 200Hz.

The Simulation environment includes LIPM simulation based on Simulink, and a self-built model with IMU and F/T sensors in gazebo, where we can set inertia parameters and other ode parameters like the real world.

We have test our new algorithm online in the robot and plan to take it to compete in the RoboCup2017, Nagoya.

A. Simulation in Gazebo

We set two levels of external force we add to the models, and each force sustained effect for 100ms to change the state of the robot and its CoM. The RMS CoP's error δu_{RMS} is shown in Table I.

$$\delta u_{RMS} = \sqrt[2]{\frac{\sum_{k=1}^{n} \delta u_{x_k}^2 + \delta u_{y_k}^2}{n}}$$

TABLE I. Experiments with Different Force Effect

$\boldsymbol{\delta u_{RMS}}$(m)	**Push Level (continues 0.1s)**		
	No Push	***50N***	***100N***
Fixed	0.0559	0.1106	NaN
Optimized	0.0427	0.0614	0.1028

In the gazebo Simulation, we can find that by optimizing the step timing, walking on a series of fixed footprints can also be robust to large disturbance. The traditional DCM controller are able to withstand some small force by shifting its CoP to the support polygon's boundary, but finally fall down when the push leads to the divergent component uncontrollable.

B. *Time adjustment in Real Hardware*

1) Stable Walking Without Push: There are some bricks with the same size as the robot's foot on the ground, we set our walking gait generator to plan a trajectory right on these fixed footholds. The Dynamics FF&FB controllers ensure our robot walking on it fluently without any external force. The results of the experiments are shown in Fig. 9. The robot walk toward 11 steps with a step length of 15cm.

2) Push Detection: We use the IMU fixed in the middle of the hip to estimate the CoM's velocity and acceleration. We use Kalman filter in the program to smooth the gyro's data. Moreover, we calculate the Position by Forward Kinematics and Odometer. When we find the estimated state has a step jump, the controller started to optimize a proper step timing to make the swing leg move faster or slower until the force disappears.

3) Robust Walking on Bricks: We Push the robot when he is walking from one brick to another one during single support phase. It can be seen from Fig. 10 that compared with fixed time walking (Fig. 10 (b)), the robot returns to normal much faster by modifying CoP and step timing(Fig. 10 (a)). What's more, step timing optimization can also help the CoP back to the desired trajectory. Fig. 11 shows the coresponding motion when the robot being pushed while walking, which can be back from tilt through step timing adjustment.

V. Conclusions And Future Works

A. *Conclusions*

In this paper, we proposed a optimization method for generating robust walking gait when the footholds are limited to the robot. And we found that when the force is parallel to the way forward or backward, it will be worth adjusting step timing to help the robot avoid falling down, other than we should use a multi objective optimization to sacrifice current step's target DCM accuracy for a sustainable robust walking gait. To reduce the computational complexity of the optimization problem, we lift up a nearly equivalence problem, which can get the approximate optimal solution as the complex one, but will decrease the time complexity of the problem to $O(n)$. Finally, our experiments in the simulations and our hardware platform proves that the effect of step timing and CoP optimization on walking stability is positive. When the force is parallel to the forward, the robot moves faster and take less time in current step, but will be walk slower to prevent the force when the robot is pulled backward.

B. *Future works*

1) CoP Controller: The CoP controller in our system has a narrow bandwidth, which is the bottlenecks of the whole motion control system. For the hardware we use should be upgrade and the algorithm need to be more powerful. We will try to use more accurate model such as whole body inverse dynamics to reduce the error between model and real robot.

2) DCM Controller in Double Support Phase: It is obvious that the DCM can be controlled more flexibly in DSP, for the support polygon now is almost three times that in single support phase. We will integrate them together in the near future.

3) Time-Varing LIP: The height of the LIP is constant all the time, while the real CoM can not strictly fixed to the middle of the hip. We will consider the variable height of the LIP into the walking gait generation.

Acknowledgment

This work is supported by the National Nature Science Foundation of China (No.61273357). We acknowledge all the reviewers' comments, which are valuable to us.

References

[1] M. Vukobratovic and D. Juricic, "Contribution to the synthesis of biped gait," *IEEE Transactions on Biomedical Engineering*, no. 1, pp. 1-6, 1969.

[2] I. Kato, S. Ohteru, H. Kobayashi, K. Shirai, and A. Uchiyama, "Information-power machine with senses and limbs," in *On Theory and Practice of Robots and Manipulators*: Springer, 1974, pp. 11-24.

[3] S. Kajita and K. Tani, "Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode," in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, IEEE, 1991, pp. 1405-1411.

[4] S. Kajita et al., "Biped walking pattern generation by using preview control of zero-moment point," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, IEEE, 2003, vol. 2, pp. 1620-1626.

[5] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, 2006, pp. 200-207: IEEE.

[6] T. Takenaka, T. Matsumoto, and T. Yoshiike, "Real time motion generation and control for biped robot-1 st report: Walking gait pattern generation," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, IEEE, 2009, pp. 1084-1091.

[7] A. L. Hof, "The 'extrapolated center of mass' concept suggests a simple control of balance in walking," *Human movement science*, vol. 27, no. 1, pp. 112-125, 2008.

[8] M. Krause, J. Englsberger, P.-B. Wieber, and C. Ott, "Stabilization of the capture point dynamics for bipedal walking based on model predictive control," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 165-171, 2012.

[9] J. Englsberger and C. Ott, "Integration of vertical COM motion and angular momentum in an extended Capture Point tracking controller for bipedal walking," in *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*, IEEE, 2012, pp. 183-189.

[10] T. Koolen, T. De Boer, J. Rebula, A. Goswami, and J. Pratt, "Capturability-based analysis and control of legged locomotion, Part 1: Theory and application to three simple gait models," *The International Journal of Robotics Research*, vol. 31, no. 9, pp. 1094-1113, 2012.

[11] R. J. Griffin, A. Leonessa, and A. Asbeck, "Disturbance compensation and step optimization for push recovery," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, IEEE, 2016, pp. 5385-5390.

[12] P. Kryczka, P. Kormushev, N. G. Tsagarakis, and D. G. Caldwell, "Online regeneration of bipedal walking gait pattern optimizing footstep placement and timing," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, IEEE, 2015, pp. 3352-3357.

[13] M. Khadiv, A. Herzog, S. A. A. Moosavian, and L. Righetti, "Step timing adjustment: A step toward generating robust gaits," in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*, IEEE, 2016, pp. 35-42.

[14] J. Englsberger, C. Ott, M. A. Roa, A. Albu-Schäffer, and G. Hirzinger, "Bipedal walking control based on capture point dynamics," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, IEEE, 2011, pp. 4420-4427.

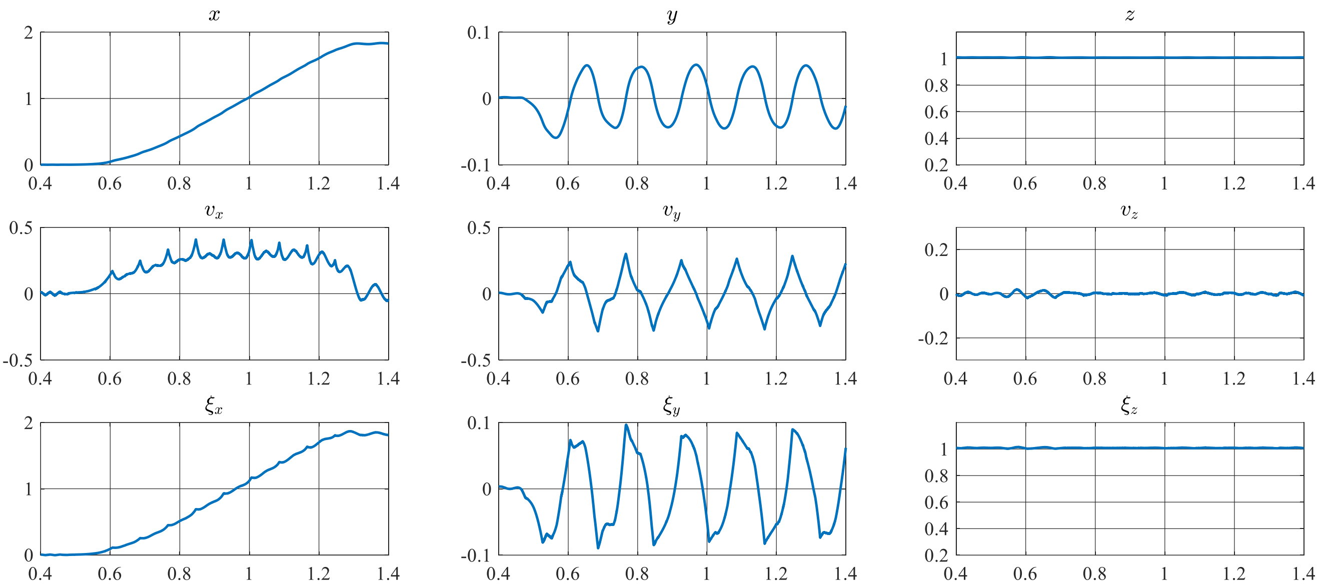

Fig. 9. Experimental results($x, \dot{x}, \xi$) for DCM controller while no external force added.

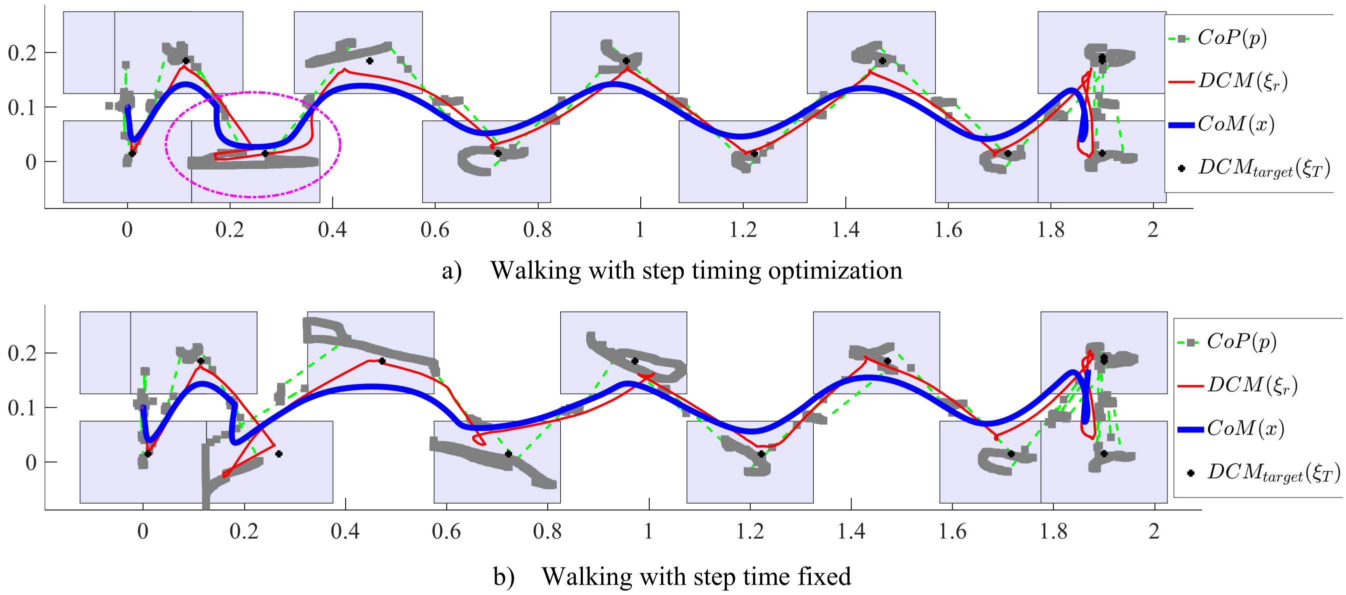

Fig. 10. CoP and DCM trajectory with and without step timing optimization

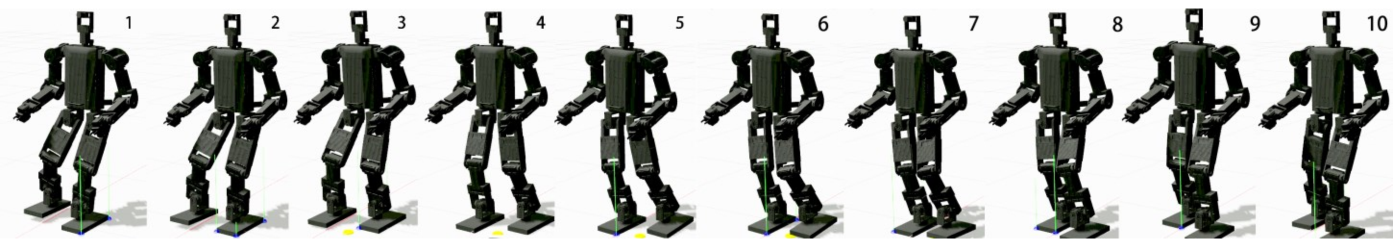

Fig. 11. Sequence of the robot walking