Research Article

INTERNATIONAL JOURNAL OF
ADVANCED ROBOTIC SYSTEMS

International Journal of Advanced
Robotic Systems
January-February 2020: 1–15
© The Author(s) 2020
DOI: 10.1177/1729881419887291
journals.sagepub.com/home/arx
SAGE

Flexible model predictive control based on multivariable online adjustment mechanism for robust gait generation

Sheng Dong[1], Zhaohui Yuan[1], Xiaojun Yu[1], Muhammad Tariq Sadiq[1], Jianrui Zhang[1], Fuli Zhang[1] and Cheng Wang[2]

Abstract

The gait generation algorithm considering both step distance adjustment and step duration adjustment could improve the anti-disturbance ability of the humanoid robot, which is very important to the dynamic balance, but the step duration adjustment often brings non-convex optimization problems. In order to avoid this situation and improve the robustness of the gait generator, a gait generation mechanism based on flexible model predictive control is proposed in this article. Specifically, the step distance adjustment and step duration adjustment are set to be optimization objectives, while the change of pressure center is treated as the optimal input to minimize those objectives. With the current system state being used for online re-optimization, a feedback gait generator is formed to realize the strong stability of variable speed and variable step distance walking of the robot. The main contributions of this work are twofold. First, a gait generation mechanism based on flexible model predictive control is proposed, which avoids the problem of nonlinear optimization. Second, a variety of feasible optimization constraints were considered, they can be used on platforms with different computing resources. Simulations are conducted to verify the effectiveness of the proposed mechanism. Results show that as compared with those considering step adjustment only, the proposed method largely improves the compensation ability of disturbance and shortens the adjustment time.

Keywords

Humanoid robots, model predictive control, gait generation, adaptive step duration

Date received: 20 May 2019; accepted: 16 October 2019

Topic: Humanoid Robotics
Topic Editor: Yannick Aoustin
Associate Editor: Olfa Boubaker

Introduction

Leg movement affects the dynamic balance of humanoid robots, and it is the basic guarantee for robots to complete various advanced tasks. However, owing to the complexities of its hybrid dynamics, the unidirectional constraints on contact forces, as well as the high dimensionality and nonlinearity of robot general dynamics, the robot leg movement is widely regarded to be a difficult problem[1] and various methods have been proposed.[1–7] Specifically, to

[1] School of Automation, Northwestern Polytechnical University, Xi'an, China

[2] School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, China

Corresponding author:
Xiaojun Yu, School of Automation, Northwestern Polytechnical University, Xi'an 710072, China.
Email: XJYU@nwpu.edu.cn

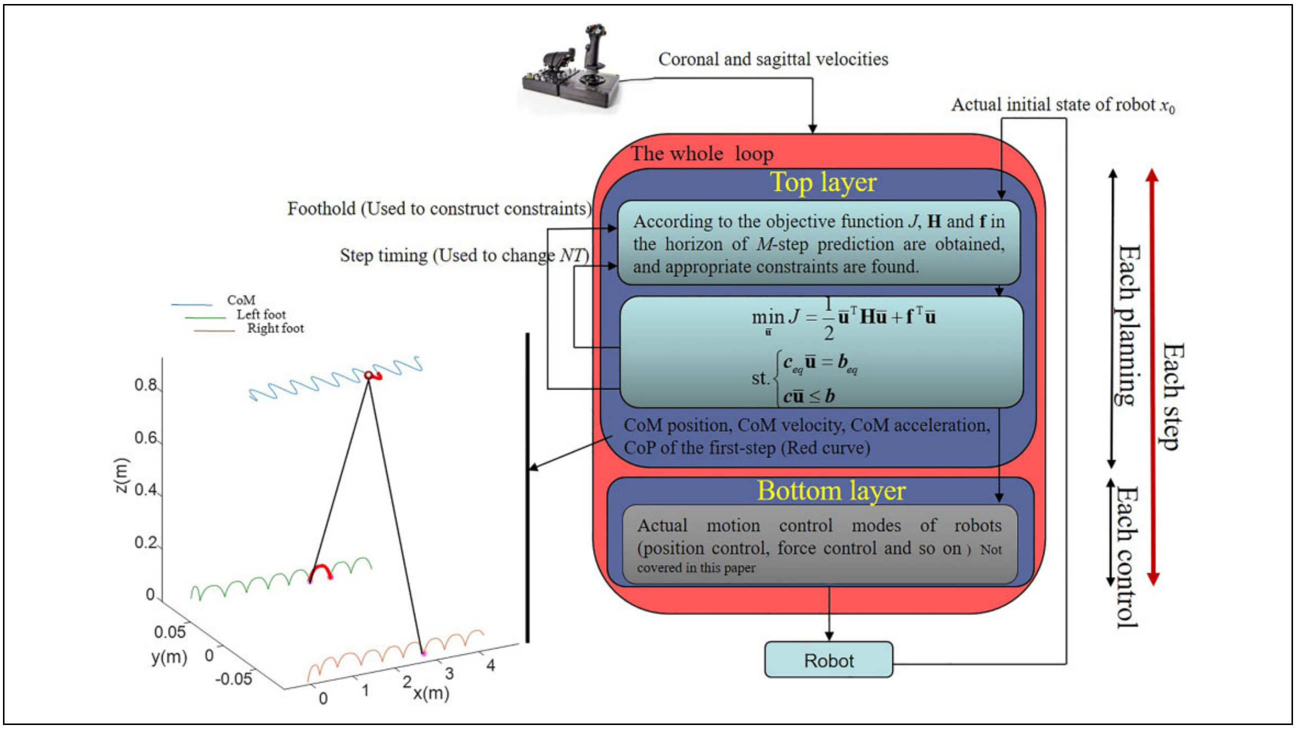
Coronal and sagittal velocities
Actual initial state of robot x_0
The whole loop
Top layer
Foothold (Used to construct constraints)
Step timing (Used to change NT)
According to the objective function J, $\mathbf{H}$ and $\mathbf{f}$ in the horizon of M-step prediction are obtained, and appropriate constraints are found.
$\min_{\bar{\mathbf{u}}} J = \frac{1}{2}\bar{\mathbf{u}}^{\mathrm{T}}\mathbf{H}\bar{\mathbf{u}} + \mathbf{f}^{\mathrm{T}}\bar{\mathbf{u}}$
st. $\begin{cases} \boldsymbol{c}_{eq}\bar{\mathbf{u}} = \boldsymbol{b}_{eq} \\ \boldsymbol{c}\bar{\mathbf{u}} \le \boldsymbol{b} \end{cases}$
CoM position, CoM velocity, CoM acceleration, CoP of the first-step (Red curve)
Bottom layer
Actual motion control modes of robots (position control, force control and so on) Not covered in this paper
Robot
Each planning
Each step
Each control
CoM
Left foot
Right foot
z(m)
0.8
0.6
0.4
0.2
0
y(m)
0.05
0
-0.05
x(m)
0
1
2
3
4

Figure 1. Block diagram of the variable step duration method. At the beginning of each step, according to the actual state and expected speed, the Hessian matrix and the constraints in M-step prediction horizon are determined. Then, through a QP, the landing point of the swing foot and the next step duration are adjusted. The key point trajectories of CoM and CoP are produced. The swing foot trajectories are interpolated according to the landing point (not covered in this article). Only the first step in the generated M-walking trajectory is used for subsequent control. QP: quadratic programming; CoM: Center of Mass; CoP: Center of Pressure.

address the dynamic balancing problem of humanoid robots, a general method[8,9] is to use a hierarchical structure shown in Figure 1 to control the robot. As illustrated, the bottom layer is a motion control layer, and it is responsible for tracking the planned robot trajectory by utilizing inverse kinematics or inverse dynamics. While the top level is the motion planning layer, which is utilized to generate a set of desired targets, for example, the Center of Mass (CoM), the Center of Pressure (CoP), the Center of Gravity (CoG),[10] limb movements as well as some other tracks. The top layer determines the walking gait of the robot and is a prerequisite for stable work. In the case of strong disturbance, it has to adjust the step of the robot quickly to the right position within a period of time like the human beings.

For top-level motion planning, many gait generation schemes perform zero moment point (ZMP, equivalent to CoP[2]) conditions by calculating the trajectory suitable for robot's CoM. Most of them use simplified CoM dynamics models, such as the famous linear inverted pendulum (LIP) model[3] and Cart-Table (CT) model.[4] Using a simplified model to capture task-related dynamic processes to a set of linear equations is very useful for real-time generation of walk patterns. Specifically, Kajita et al.[4] put forward the concept of preview control and solved a linear quadratic (LQ) regulation problem for a dynamical extension of the CT. Wieber[5] further developed it into model predictive control (MPC) and selected the CoM jerk as control signal to achieve asymptotic tracking of the desired ZMP trajectory. However, both methods are not strong enough to disturbance. Recently, Englsberger et al.[6] and Takenaka et al.[7] split the LIP model into a stable first-order system and an unstable first-order system, and then introduced the intermediate variables of the divergence component of motion (DCM or capture point[11]) to solve the CoP and CoM trajectories online. Within these methods, the input of the unstable subsystem in LIP is planned online according to the predetermined footstep locations, while both the CoP and CoM trajectories were generated in real time by tracking the DCM according to the robot natural dynamics. Englsberger et al.[12] extended the method to 3-D case. Yet, it is worth noting that although these methods have certain resistance to perturbations, their effects are still limited.

To enhance the robustness of the generated gait, Herdt et al.[13] and Diedam et al.[2] used predictive control to include step distance to the optimization objective function, while made footsteps variable to generate both CoM and CoP trajectories under the CoP constraints. Results showed that the generated walking mode is robust to disturbances and achieves high level target tasks, such as desired step position or walking speed. However, the step distance adjustment contributes only to disturbance energy consumption, and they do not take into account the step timing to change the landing speed either. In practice, step duration adjustment could help adjust the robot foot movement speed for more robust gait generation. Since the CoM trajectory is a nonlinear function of time and of initial conditions, in order to deal with nonlinear optimization problems, many algorithms have been proposed in the literature.[14–17] Kryczka et al.[18] and Aftab et al.[19] proposed to utilize a nonlinear optimization technique to modify both footstep positions and step-timing to maintain dynamic stability during the robot walking process. Specifically, Aftab et al. introduced a simple model of the mechanical cost in the objective function by penalizing the acceleration of the swing foot, while the acceleration can't anyway exceed a given maximum value. Kryczka et al. rewrote the optimization objective to be a nonlinear function of the optimization variables. Although satisfactory results have been achieved, the nonlinear optimization process introduced high computational costs and cannot guarantee convergence to the minimum either.

Khadiv et al.[20] adopted a DCM method by taking the constraints of both location and time of stepping into account in robot gait generation process and modeled the problem as a quadratic program that can be solved real time. The results showed that such the proposed strategy could help the robot recover from severe pushes. However, since the CoM motion constraint has not been taken into account in the objective function, the CoM speed tracking error cannot be minimized in the control process. Sun et al.[21] proposed a class of global and feasible projected Fletcher–Reeves conjugate gradient approach. This method guarantees the tracking accuracy of each physical quantity to the target value in the optimization process but requires the optimization constraints to be linear. Sun et al.[22] further proposed a superlinearly convergent trust region-sequential quadratic programming (QP) approach. The method incorporates a combination algorithm that allows both the trust region technique[23,24] and the sequential QP method to be used. It avoids solving the QP subproblem for nonlinear constrained optimization problems, which gives the potential for fast convergence in the neighborhood of an optimal solution.

In this article, we propose to flexibly choose the analytical bounds of position, velocity, and acceleration of CoM in MPC frame to predict the stability and manually set the target footstep locations and step duration in the prediction horizon. The optimal outputs are allowed to deviate from the target, that is, to adjust the item weight coefficient in the optimization objective, and the variation of CoP are selected as the input to minimize the objective. With the current system state being used to recalculate the optimization online, a feedback controller as shown in Figure 1 is formed, which outputs the required CoM, CoP, footstep locations, and the next step timing. Finally, according to the practical robot structure, the maximum and minimum stride distances of the lateral plane and the sagittal plane are also included as the optimization constraint of QP to achieve variable speed and variable step.

Table 1. The full name of acronyms.

The acronyms	The full name
MPC	Model predictive control
CoM	Center of Mass
CoP	Center of Pressure
CoG	Center of Gravity
DCM	Divergence component of motion
LIP	Linear inverted pendulum
LQ	Linear quadratic
QP	Quadratic programming
ZMP	Zero moment point
CT	Cart-Table

The main contributions of this study could be summarized as follows. First, we improved the MPC objective function with adjustment of step duration taken into account, and therefore, such a step timing adjustment strategy could largely suppress the interference in forward visual field and stabilize CoM velocity mutation with the non-convex optimization problem avoided, as compared with those in the existing work.[2,4,25] Second, the proposed method flexibly handles the stability optimization problem for biped robots and also presents a variety of gait optimization constraints, which can be easily implemented on control platforms with different computing resources. With the foot CoP stabilized without any mutations in the forward motion, the proposed mechanism mimics human walking, and thus is conducive to walking smoothly with small errors.

The article is organized as follows. In the "The MPC method for walking machine" section, we recall the LIP and the CoM trajectory characterization of standard MPC. The "System optimization model" section describes the motion model adopted, the "Walking planning with adjustable step duration" section achieves flexible walking optimization through carefully selected objective function and constraints, and the "Push recovery planning" section describes push recovery. To illustrate its benefits, all the simulation results are shown in "Results and discussion" section, the conclusion points out the next research work. For simplicity purpose, the abbreviations utilized in this article are summarized in Table 1 as below.

The MPC method for walking machine

When a robot is walking on the ground, it comes down to the fact that the CoP can lie only within the convex hull of the contact points between the robot's feet and the ground. For the LIP model used by most scholars, it describes the motion of the CoM of a robot when its height is constant while the rotation effect is not considered. In addition to its linear properties, the LIP has the advantage that dynamics along the x- and y-directions are decoupled and can be represented by the same differential equations. Many studies[4,26–30] have proved that, despite its simplicity, the use of

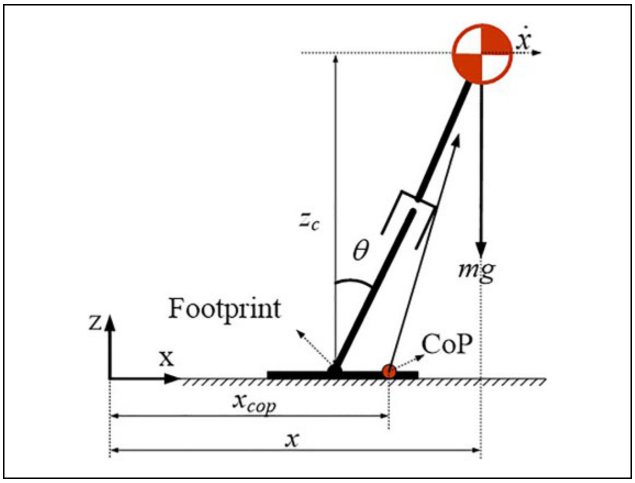

Figure 2. The LIP model of biped. LIP: linear inverted pendulum.

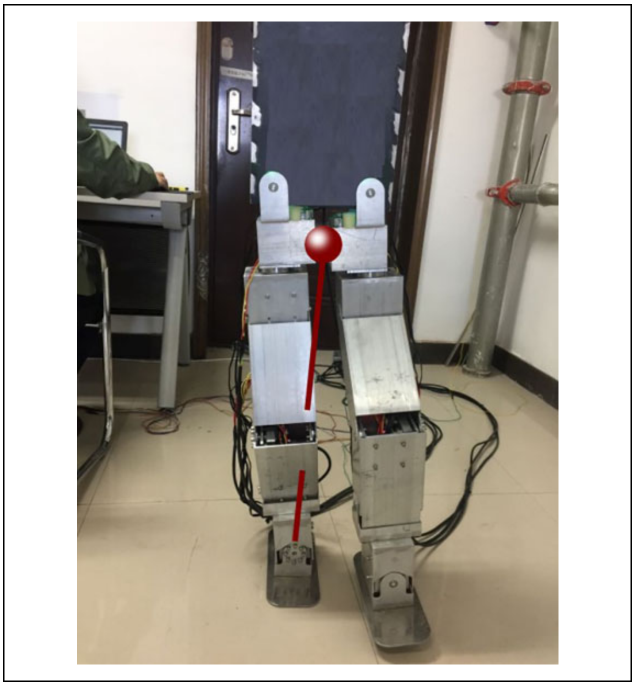

Figure 3. The biped robot.

LIP (see Figure 2) for gait generation is effective. Simplified LIP model for real robots is shown in Figure 3. Establish LIP dynamic equation in the x-axis (the y-axis identical)

$$\ddot{x} = \omega^2(x - x_{\text{cop}}) \tag{1}$$

where $\omega = \sqrt{g/z_c}$, with g and z_c being the acceleration of gravity and the constant altitude of CoM, respectively. x is the horizontal position of the CoM, $\ddot{x}$ is horizontal acceleration of the CoM, and x_{cop} is the position of the CoP.

It could be found in equation (1) that LIP has a right half plane pole, and there is always a divergence component in the system, which leads to the divergence of CoM. In addition, the change of CoP will directly lead to the change of the divergence velocity of CoM. However, CoP, the input signal has some constraints and must be within the range of stable polygons. The scheme proposed by Kajita et al.[4] generates a trajectory of the CoM under the constraint that the footsteps are fixed and impossible to change. CoM and CoP are programmed as a series of discrete points, with varied jerks $\dddot{x}$ over time intervals of constant lengths T. Therefore, the dynamics at $t_{k+1} = (k+1)T$ could be calculated as

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\dddot{x} \\ \boldsymbol{y}_{k+1} &= \boldsymbol{C}\boldsymbol{x}_k \end{aligned} \tag{2}$$

where $\boldsymbol{x} = [x, \dot{x}, \ddot{x}]^T$, $\boldsymbol{y} = x_{\text{cop}}$, and

$$\boldsymbol{A} = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & \mathrm{T} \\ 0 & 0 & 1 \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} T^3/6 \\ T^2/2 \\ T \end{bmatrix}$$

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & -1/\omega^2 \end{bmatrix}$$

This scheme is to minimize this jerk while maintaining a position of the CoP as close as possible to some prescribed reference positions, but in the process, there is no consideration of the robot's foothold and step timing. Therefore, to address the problems of anti-disturbance and variable speed walking for the robot, we propose to add the velocity deviation and acceleration deviation of CoM, as well as landing point deviation and stepping time deviation to the optimization function, and select the variation of CoP as the optimal input. After the first QP solution is finished, the feedback value of the system state is recalculated online to form a feedback controller, and the optimal trajectory is constantly updated.

Gait generation strategy

System optimization model

For all of the implementations presented, the model is the LIP with 2-D dynamics given by equation (1). However, the math is generally independent of the model. It is straightforward to implement any other linear model. As shown in Figure 4, a complete step includes double support phase t_{double} and single support phase t_{single}, and NT time span may contain M steps. With the state space equation of the discrete linear system shown in equation (3) taken into account, the control input u is the variation of x_{cop}. Given a sequence of control inputs $\bar{\boldsymbol{u}} = [0, u_k, u_{k+1}, \ldots, u_{k+N-1}]^T$, the $\boldsymbol{X}= [\boldsymbol{x_k}^{\boldsymbol{T}}, \boldsymbol{x}_{k+1}{}^{\boldsymbol{T}}, \boldsymbol{x}_{k+2}{}^{\boldsymbol{T}}, \ldots, \boldsymbol{x}_{k+N}{}^{\boldsymbol{T}}]^{\boldsymbol{T}}$ and $\boldsymbol{Y}= [\boldsymbol{y}_{k+1}{}^{\boldsymbol{T}}, \mathbf{y}_{k+2}{}^{T}, \ldots, \boldsymbol{y}_{k+N+1}{}^{\boldsymbol{T}}]^{\boldsymbol{T}}$ sequences within NT time intervals can be calculated

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}u_k \\ \boldsymbol{y}_{k+1} &= \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{D}u_k \end{aligned} \tag{3}$$

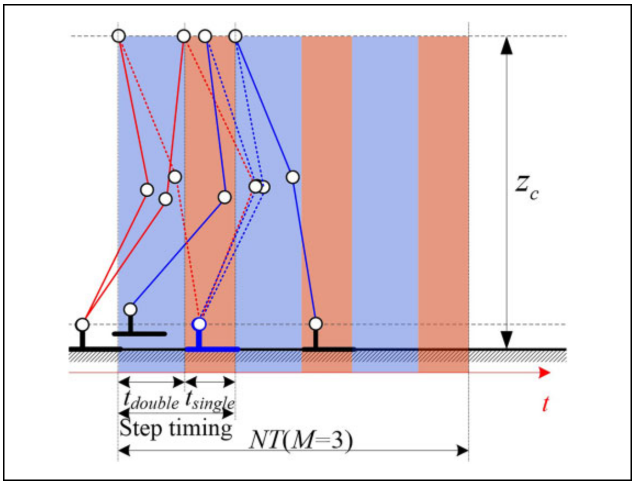

Figure 4. Predicted horizon of *M*-steps.

$$\boldsymbol{A}=\begin{bmatrix} 1+\frac{1}{2}T^2\omega^2 & T & -\frac{1}{2}T^2\omega^2 & 0 & 0 \\ T\omega^2 & 1 & T\omega^2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{B}=\begin{bmatrix} -\frac{1}{2}T^2\omega^2 \\ -T/\omega^2 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\boldsymbol{C}=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \omega^2 & 0 & -\omega^2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{D}=0$$

where $\boldsymbol{x}=[x,\dot{x},x_{\text{cop}},x_{\text{fl}},x_{\text{fr}}]^T$, $\boldsymbol{y}=[x,\dot{x},\ddot{x},x_{\text{cop}},x_{\text{fl}},x_{\text{fr}}]^T$, $x_{\text{fl}},x_{\text{fr}}$ are the left and right foot position

$$\begin{aligned} \boldsymbol{X} &= \bar{\boldsymbol{A}}\boldsymbol{x}_k+\bar{\boldsymbol{B}}\bar{\boldsymbol{u}} \\ \boldsymbol{Y} &= \bar{\boldsymbol{C}}\boldsymbol{x}_k+\bar{\boldsymbol{D}}\bar{\boldsymbol{u}} \end{aligned} \tag{4}$$

$$\bar{A}=\begin{bmatrix} I_{5*5} \\ A \\ A^2 \\ A^3 \\ \vdots \\ A^N \end{bmatrix}, \bar{B}=\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ B & 0 & 0 & 0 & 0 \\ AB & B & 0 & 0 & \vdots \\ \vdots & \vdots & \ddots & \vdots & 0 \\ A^{N-2}B & \cdots & AB & B & 0 \\ A^{N-1}B & A^{N-2}B & \cdots & AB & B \end{bmatrix}$$

$$\bar{C}=\begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N-1} \end{bmatrix}, \bar{D}=\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ CB & 0 & 0 & 0 & 0 \\ CAB & CB & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 \\ CA^{N-2}B & \cdots & CAB & CB & 0 \\ CA^{N-1}B & CA^{N-2}B & \cdots & CAB & CB \end{bmatrix}$$

In order to determine the best trajectory, a cost function is constructed. The function is a scalar function that scores the trajectory, by defining a quadratic cost on the elements of $\boldsymbol{X}$ and $\bar{\boldsymbol{u}}$ such that

$$\boldsymbol{J}=\frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{X}^T\boldsymbol{Q}\boldsymbol{X}+\frac{\mathbf{1}}{\mathbf{2}}\bar{\boldsymbol{u}}^T R\bar{\boldsymbol{u}} \tag{5}$$

Introducing it into equation (4), then

$$J=\frac{\mathbf{1}}{\mathbf{2}}\bar{\boldsymbol{u}}^T(\boldsymbol{R}+\bar{\boldsymbol{B}}^T\boldsymbol{Q}\bar{\boldsymbol{B}})\bar{\boldsymbol{u}}+\boldsymbol{x}_k^T\bar{\boldsymbol{A}}^T\boldsymbol{Q}\bar{\boldsymbol{B}}\bar{\boldsymbol{u}}+\frac{1}{2}\boldsymbol{x}_k^T\bar{\boldsymbol{A}}^T\boldsymbol{Q}\bar{\boldsymbol{A}}\boldsymbol{x}_k \tag{6}$$

It has a simple quadratic form

$$J=\frac{\mathbf{1}}{\mathbf{2}}\bar{\boldsymbol{u}}^T\boldsymbol{H}\bar{\boldsymbol{u}}+\boldsymbol{f}\bar{\boldsymbol{u}}+J_0 \tag{7}$$

where

$$\begin{aligned} \boldsymbol{H} &= \boldsymbol{R}+\bar{\boldsymbol{B}}^T\boldsymbol{Q}\bar{\boldsymbol{B}}, \\ \boldsymbol{f} &= \boldsymbol{x}_k^T\bar{\boldsymbol{A}}^T\boldsymbol{Q}\bar{\boldsymbol{B}}, \\ J_0 &= \frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{x}_k^T\bar{\boldsymbol{A}}^T\boldsymbol{Q}\bar{\boldsymbol{A}}\boldsymbol{x}_k \end{aligned}$$

Note that the J_0 term is constant with respect to $\bar{\mathbf{u}}$ and has no effect on the location of the minimum of J.

Adding equality and inequality dynamic equilibrium constraints

$$\boldsymbol{c}_{\boldsymbol{eq}}\bar{\boldsymbol{u}}=\boldsymbol{b}_{\boldsymbol{eq}} \tag{8}$$

$$\boldsymbol{c}_{\boldsymbol{in}}\bar{\boldsymbol{u}}\leq\boldsymbol{b}_{\boldsymbol{in}} \tag{9}$$

For some constrained states, $\boldsymbol{c}'_{\boldsymbol{in}}\boldsymbol{X}\leq\boldsymbol{b}'_{\boldsymbol{in}}$ can be transformed into

$$\boldsymbol{c}'_{\boldsymbol{in}}\bar{\boldsymbol{B}}\bar{\boldsymbol{u}}\leq\boldsymbol{b}'_{\boldsymbol{in}}-\boldsymbol{c}'_{\boldsymbol{in}}\bar{\boldsymbol{A}}\boldsymbol{x}_k \tag{10}$$

Finally, the problem of finding CoM trajectory in a period of time NT is converted into a QP. There are many methods can be used to solve this standard QP, such as interior points, active sets, conjugant gradient, or simplex methods[31]

$$\min_{\bar{u}} J=\frac{\mathbf{1}}{\mathbf{2}}\bar{\boldsymbol{u}}^T\boldsymbol{H}\bar{\boldsymbol{u}}+\boldsymbol{f}\bar{\boldsymbol{u}},$$

$$\text{st.}\begin{cases} \boldsymbol{c}_{\boldsymbol{eq}}\bar{\boldsymbol{u}}=\boldsymbol{b}_{\boldsymbol{eq}} \\ \begin{bmatrix} \boldsymbol{c}_{\boldsymbol{in}} \\ \boldsymbol{c}'_{\boldsymbol{in}}\bar{\boldsymbol{B}} \end{bmatrix}\bar{\boldsymbol{u}}\leq\begin{bmatrix} \boldsymbol{b}_{\boldsymbol{in}} \\ \boldsymbol{b}'_{\boldsymbol{in}}-\boldsymbol{c}'_{\boldsymbol{in}}\bar{\boldsymbol{A}}\boldsymbol{x}_k \end{bmatrix} \end{cases} \tag{11}$$

Walking planning with adjustable step duration

The MPC scheme adopted by Kajita et al.[4] is basic in generating stable CoM trajectories. Its only mandatory feature is to adjust the magnitude of the motion derivative of CoM in the prediction horizon. However, any control variables that help to suppress CoM mutations contribute to more stable walking.

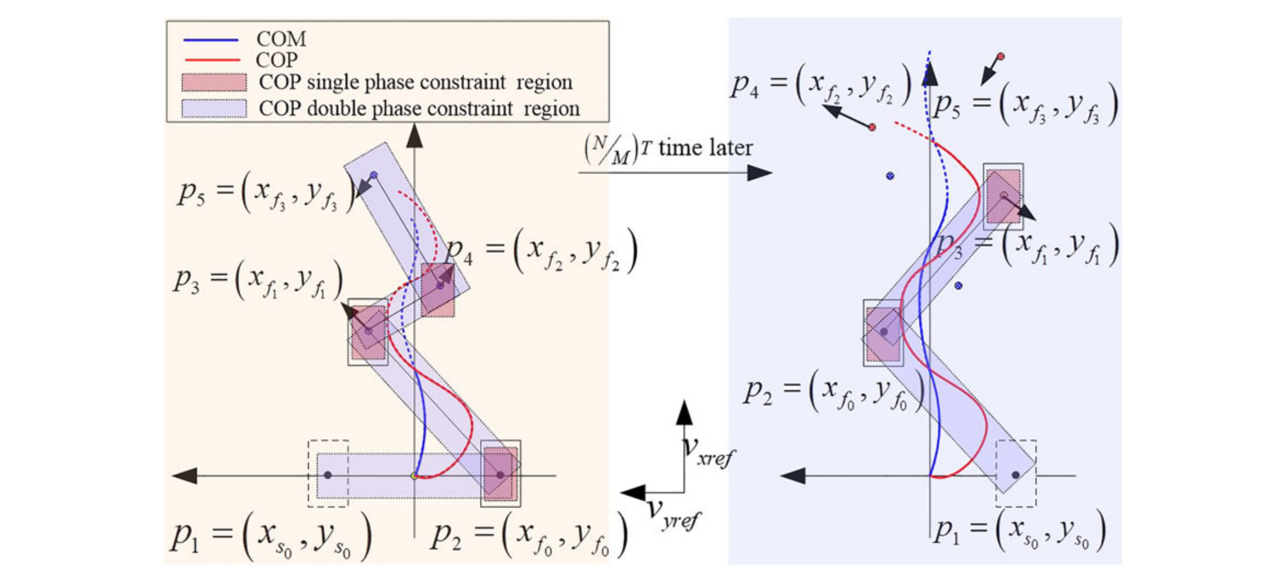
COM
COP
COP single phase constraint region
COP double phase constraint region
$(N/M)T$ time later
$p_5 = (x_{f_3}, y_{f_3})$
$p_4 = (x_{f_2}, y_{f_2})$
$p_3 = (x_{f_1}, y_{f_1})$
$p_2 = (x_{f_0}, y_{f_0})$
$p_1 = (x_{s_0}, y_{s_0})$
v_{xref}
v_{yref}

Figure 5. *M*-steps quadratic programming process.

Objective function. For the dynamic balance of biped robot, the form of function will be different under different circumstances. In case of steady walking, this article takes the following form of objective function

$$\begin{aligned} J = \; & \frac{a_1}{2} \| \boldsymbol{C}_x^{\text{target}} - \boldsymbol{C}_x \|^2 + \frac{a_2}{2} \| \dot{\boldsymbol{C}}_x^{\text{target}} - \dot{\boldsymbol{C}}_x \|^2 \\ & + \frac{a_3}{2} \| \ddot{\boldsymbol{C}}_x^{\text{target}} - \ddot{\boldsymbol{C}}_x \|^2 + \frac{a_4}{2} \| \bar{u}_x \|^2 \\ & + \frac{a_5}{2} \| x_f^{\text{target}} - x_f \|^2 + \frac{a_6}{2} \| T_{x\text{step}}^{\text{target}} - T_{x\text{step}} \|^2 \\ & + \frac{a_1}{2} \| \boldsymbol{C}_y^{\text{target}} - \boldsymbol{C}_y \|^2 + \frac{a_2}{2} \| \dot{\boldsymbol{C}}_y^{\text{target}} - \dot{\boldsymbol{C}}_y \|^2 \\ & + \frac{a_3}{2} \| \ddot{\boldsymbol{C}}_y^{\text{target}} - \ddot{\boldsymbol{C}}_y \|^2 + \frac{a_4}{2} \| \bar{u}_y \|^2 \\ & + \frac{a_5}{2} \| y_f^{\text{target}} - y_f \|^2 + \frac{a_6}{2} \| T_{y\text{step}}^{\text{target}} - T_{y\text{step}} \|^2 \end{aligned} \tag{12}$$

where $(\boldsymbol{C}_x, \boldsymbol{C}_y)$, $(\dot{\boldsymbol{C}}_x, \dot{\boldsymbol{C}}_y)$, and $(\ddot{\boldsymbol{C}}_x, \ddot{\boldsymbol{C}}_y)$ are the vectors of CoM position, velocity, and acceleration in 2-D over the next N time steps, respectively. $(\boldsymbol{C}_x^{\text{target}}, \boldsymbol{C}_y^{\text{target}})$, $(\dot{\boldsymbol{C}}_x^{\text{target}}, \dot{\boldsymbol{C}}_y^{\text{target}})$, and $(\ddot{\boldsymbol{C}}_x^{\text{target}}, \ddot{\boldsymbol{C}}_x^{\text{target}})$ are the target position, target velocity, and target acceleration of CoM, respectively. $(\bar{\boldsymbol{u}}_x, \bar{\boldsymbol{u}}_y)$ in this case is the variation of the x_{cop} in 2-D. $\boldsymbol{x}_f^{\text{target}}, \boldsymbol{y}_f^{\text{target}}, \boldsymbol{x}_f$, and y_f are vectors of the next M reference and actual foothold, respectively. $T_{x\text{step}}^{\text{target}}, T_{y\text{step}}^{\text{target}}$ are the target step timing in the x-direction and in the y-direction, respectively. $T_{x\text{step}}$ and $T_{y\text{step}}$ are the optimal outputs in their respective directions, a_1, a_2, a_3, a_4, a_5, and a_6 are the weight coefficients of corresponding variables in J.

As shown in Figure 5, in the calculation at a time, the robot can walk one step or walk $M > 1$ steps. So here, we chose to give it three steps, $M = 3$, where $x_f = [x_{f_1}, x_{f_2}, x_{f_3}]^T$, $y_f = [y_{f_1}, y_{f_2}, y_{f_3}]^T$. The reference value is selected as

$$\begin{cases} \boldsymbol{C}_x^{\text{target}} = \frac{1}{2}(x_{s_0} + x_{f_0}) + v_{x\text{ref}} NT \\ \boldsymbol{C}_y^{\text{target}} = \frac{1}{2}(y_{s_0} + y_{f_0}) + v_{y\text{ref}} NT \end{cases},$$

$$\begin{cases} \dot{\boldsymbol{C}}_x^{\text{target}} = v_{x\text{ref}} \\ \dot{\boldsymbol{C}}_y^{\text{target}} = v_{y\text{ref}} \end{cases}, \begin{cases} \ddot{\boldsymbol{C}}_x^{\text{target}} = 0 \\ \ddot{\boldsymbol{C}}_y^{\text{target}} = 0 \end{cases}, \begin{cases} T_{x\text{step}}^{\text{target}} = t_{\text{step}} \\ T_{y\text{step}}^{\text{target}} = t_{\text{step}} \end{cases}$$

$$\begin{cases} \boldsymbol{x}_f^{\text{target}} = \left[x_{f_0} + \frac{1}{2} v_{x\text{ref}}, x_{f_1} + \frac{1}{2} v_{x\text{ref}}, x_{f_2} + \frac{1}{2} v_{x\text{ref}} \right]^T \\ \boldsymbol{y}_f^{\text{target}} = [y_{f_0} + d, y_{f_1} - d, y_{f_2} + d] \end{cases}$$

$d = 0.2\text{sign}(y_{s_0} - y_{f_0})$, (x_{s_0}, y_{s_0}) is the position of initial swing foot, (x_{f_0}, y_{f_0}) is the position of initial support foot, $v_{x\text{ref}}$ and $v_{y\text{ref}}$ are the reference velocity of CoM, t_{step} is a known deterministic value. Note: The $(\boldsymbol{x}_f, \boldsymbol{y}_f)$, $T_{x\text{step}}$, and $T_{y\text{step}}$ are unknown in advance, the objective function J can be written as a function of $\boldsymbol{u}$

$$\boldsymbol{u} = [\bar{\boldsymbol{u}}_x^T, x_f^T, T_{x\text{step}}, \bar{\boldsymbol{u}}_y^T, y_f^T, T_{y\text{step}}]^T \in R^{2(N+M+1)} \tag{13}$$

After solving the QP, the value of $\boldsymbol{u}$ is obtained. The CoM and CoP trajectories can be got from equation (4) and initial status x_0. Take the minimum values in $T_{x\text{step}}$ and $T_{y\text{step}}$ for the next QP, that is, $T = \min(T_{x\text{step}}, T_{y\text{step}})$. Although we worked out the CoM and CoP trajectories in the M-walk process, only the first foothold is taken as the current destination, the QP is recomputed in next NT time.

Constraints. Constraints play a major role in the computation of the optimal trajectory. In the robot walking process, CoP directly determines the balance state of the robot. It should also be noted that since the locations of the footsteps are variables in optimization process, the constraints on the CoP after the first step are unknown, and thus, it is necessary to dynamically set CoP constraints according to the optimized planning footstep locations. In addition, in order to maintain a simple linear form and use a fast QP solver, simple conservative constraints are chosen to approximate the real constraints. The conservative constraints represent smaller regions than the real constraints. Another advantage of this approach is that it introduces a safety margin and enhances the stability of optimization.

- CoP constraints

Since robot is in a different supporting phase at different times, the CoP constraints could be divided into a set Z^S_{cop} consisting of multiple single support constraint blocks and a set Z^D_{cop} consisting of multiple double support constraint blocks throughout the NT time (see Figure 5).

Single support phase (during t_{single})

When in single support, the CoP can only be within the range of the support foot, so the feasible range of the CoP is

$$\begin{aligned} \sum_{i=0}^{2}(x_{f_i-\Delta_1})\boldsymbol{U}^S_i \le \boldsymbol{x_{cop}} \le \sum_{i=0}^{2}(x_{f_i+\Delta_2})\boldsymbol{U}^S_i \\ \sum_{i=0}^{2}(y_{f_i-\Delta_3})\boldsymbol{U}^S_i \le y_{cop} \le \sum_{i=0}^{2}(y_{f_i+\Delta_4})\boldsymbol{U}^S_i \end{aligned} \tag{14}$$

where $\boldsymbol{U}^S_i \in R^{N\times 1}$ be defined as a vector of zeros and ones with the ones corresponding to the time steps of the ith single phase, with the first phase being the initial double support phase. $\Delta_1, \Delta_2, \Delta_3$, and Δ_4 are the stable ranges, which must be less than half of the foot size. From equation (4), we get

$$\begin{aligned} \boldsymbol{x_{cop}} = \bar{\boldsymbol{A}}_{\boldsymbol{x_{cop}}}\boldsymbol{x_k} + \bar{\boldsymbol{B}}_{\boldsymbol{x_{cop}}}\bar{\boldsymbol{u}} \\ \boldsymbol{y_{cop}} = \bar{\boldsymbol{A}}_{\boldsymbol{y_{cop}}}\boldsymbol{x_k} + \bar{\boldsymbol{B}}_{\boldsymbol{y_{cop}}}\bar{\boldsymbol{u}} \end{aligned} \tag{15}$$

$\bar{\boldsymbol{A}}_{\boldsymbol{x_{cop}}}, \bar{\boldsymbol{B}}_{\boldsymbol{x_{cop}}}$ and $\bar{\boldsymbol{A}}_{\boldsymbol{y_{cop}}}, \bar{\boldsymbol{B}}_{\boldsymbol{y_{cop}}}$ are the corresponding line in $\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}}$. Finally, we have

$$\begin{aligned} \sum_{i=0}^{2}(x_{f_i-\Delta_1})\boldsymbol{U}^S_i \le \bar{\boldsymbol{A}}_{\boldsymbol{x_{cop}}}x_k + \bar{\boldsymbol{B}}_{\boldsymbol{x_{cop}}}\bar{\boldsymbol{u}} \le \sum_{i=0}^{2}(x_{f_i+\Delta_2})\boldsymbol{U}^S_i \\ \sum_{i=0}^{2}(y_{f_i-\Delta_3})\boldsymbol{U}^S_i \le \bar{\boldsymbol{A}}_{\boldsymbol{y_{cop}}}x_k + \bar{\boldsymbol{B}}_{\boldsymbol{y_{cop}}}\bar{\boldsymbol{u}} \le \sum_{i=0}^{2}(y_{f_i+\Delta_4})\boldsymbol{U}^S_i \end{aligned} \tag{16}$$

Double support phase (during t_{double})

When in double support, establish the coordinate system o' as shown in Figure 6(a) in the middle of the feet. o' rotates θ around the z-axis relative to o. Hence, for any CoP $(x_{\text{cop}}, y_{\text{cop}})$ in planning trajectory within coordinate o', it is $(x'_{\text{cop}}, y'_{\text{cop}})$, and the constraints of double support are as follows

$$\begin{aligned} \left\|\overleftarrow{x'_{\text{cop}}}\right\| \le \Delta'_x \\ \left\|\overleftarrow{y'_{\text{cop}}}\right\| \le \Delta'_y \end{aligned} \tag{17}$$

where Δ'_x, Δ'_y are the stability margin in the direction of x' and y' as illustrated in Figure 6(a), and

$$\begin{aligned} x'_{\text{cop}} &= (x_{\text{cop}} - x_r)\cos\theta - (y_{\text{cop}} - y_r)\sin\theta \\ y'_{\text{cop}} &= (y_{\text{cop}} - y_r)\cos\theta - (x_{\text{cop}} - x_r)\sin\theta - \frac{l_{\text{pace}}}{2} \end{aligned} \tag{18}$$

Because $l_{\text{pace}} = \sqrt{(x_l - x_r)^2 + (y_l - y_r)^2}$, let $l' = 1/l_{\text{pace}}$, the equation (18) can be converted to

$$\begin{aligned} x'_{\text{cop}} &= l'[(x_{\text{cop}} - x_r)(y_l - y_r) - (y_{\text{cop}} - y_r)(x_l - x_r)] \\ y'_{\text{cop}} &= l'[(y_{\text{cop}} - y_r)(y_l - y_r) - (x_{\text{cop}} - x_r)(x_l - x_r)] - \frac{1}{2l'} \end{aligned} \tag{19}$$

Then combine equation (15) and (19), equation (17) are transformed into

$$\begin{aligned} &\sum_{i=0}^{3} - l_{\text{pace}}\Delta'_x U^D_i \le [(\bar{\boldsymbol{A}}_{\boldsymbol{x_{cop}}}\boldsymbol{x_k} + \bar{\boldsymbol{B}}_{\boldsymbol{x_{cop}}}\bar{\boldsymbol{u}} - x_r\boldsymbol{U_i}^D)(y_l - y_r) \\ &\quad - (\bar{\boldsymbol{A}}_{\boldsymbol{y_{cop}}}\boldsymbol{x_k} + \bar{\boldsymbol{B}}_{\boldsymbol{y_{cop}}}\bar{\boldsymbol{u}} - y_r\boldsymbol{U_i}^D)(x_l - x_r)] \le \sum_{i=0}^{3} l_{\text{pace}}\Delta'{}_x U^D_i \\ &\sum_{i=0}^{3} - l_{\text{pace}}\Delta'_y U^D_i \le [(\bar{\boldsymbol{A}}_{\boldsymbol{y_{cop}}}\boldsymbol{x_k} + \bar{\boldsymbol{B}}_{\boldsymbol{y_{cop}}}\bar{\boldsymbol{u}} - y_r\boldsymbol{U_i}^D)(y_l - y_r) \\ &\quad + (\bar{\boldsymbol{A}}_{\boldsymbol{x_{cop}}}\boldsymbol{x_k} + \bar{\boldsymbol{B}}_{\boldsymbol{x_{cop}}}\bar{u} - x_r\boldsymbol{U_i}^D)(x_l - x_r)] - \frac{l^2_{\text{pace}}}{2}U^D_i \\ &\quad \le \sum_{i=0}^{3} l_{\text{pace}}\Delta'{}_y U^D_i \end{aligned} \tag{20}$$

where $\boldsymbol{U}^D_i \in R^{N\times 1}$ be defined as a vector of zeros and ones with the ones corresponding to the time steps of the ith double phase, Hence

$$\sum_{i=0}^{2}(\boldsymbol{U}^S_i + \boldsymbol{U}^D_i) + \boldsymbol{U}^D_3 = \boldsymbol{I}_{N\times 1} \tag{21}$$

$\boldsymbol{I}_{N\times 1}$ is a N-dimensional vector with all elements being 1.

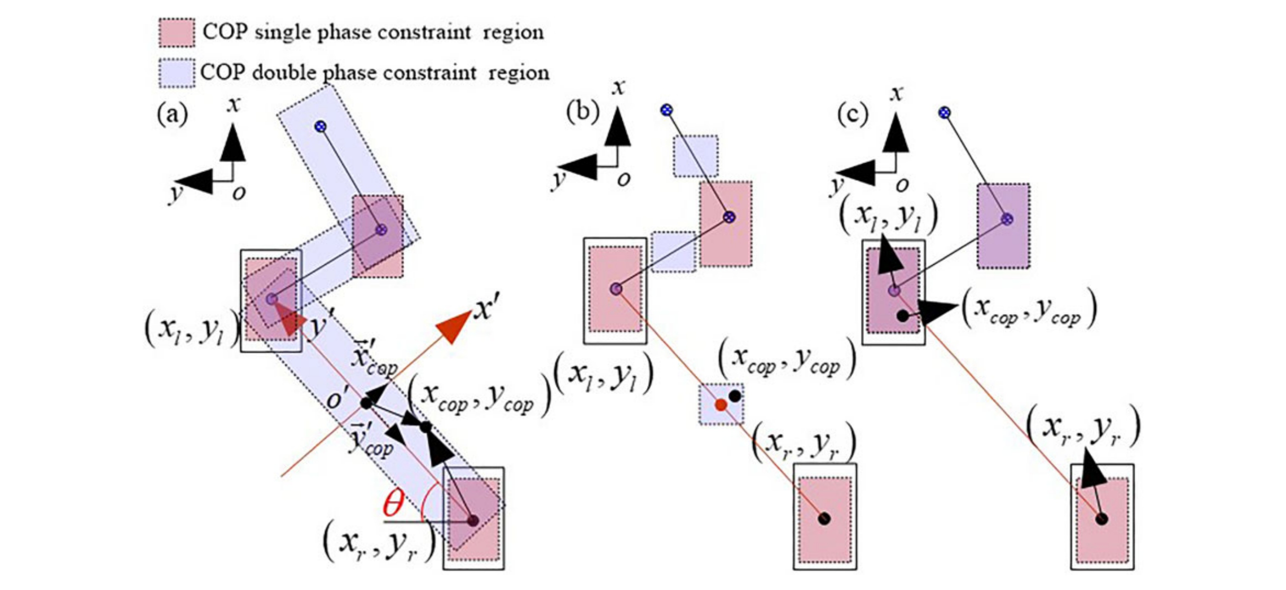

Figure 6. CoP constraints in the double support. (a) The first form, which is closest to human beings, (b) the second form, which is conservative and simplified, and (c) the third form, which is similar to fast walking. CoP: Center of Pressure.

The constraints of CoP in double support can also be selected in other forms. The optimal results obtained in different forms are slightly different in the macroscopical attitude of the robot. For example (as shown in Figure 6(b))

$$\begin{aligned} -\Delta_x \le \frac{x_l + x_r}{2} - x_{\text{cop}} \le \Delta_x \\ -\Delta_y \le \frac{y_l + y_r}{2} - y_{\text{cop}} \le \Delta_y \end{aligned} \tag{22}$$

Alternatively, the CoP constraints of the double support and the single support can be directly set up to coincide with each other as shown in Figure 6(c).

- Step distance constraints

Because of the limitation of mechanical structure, the step distance needs to be constrained

$$\begin{aligned} -\Delta_{dx} \le x_{f_i} - x_{f_{i-1}} \le \Delta_{dx} \\ -\Delta_{dy} \le y_{f_i} - y_{f_{i-1}} \le \Delta_{dy} \end{aligned} \tag{23}$$

where Δ_{dx}, Δ_{dy} are the maximum step distances of the robot.

- Foot placement constraints

When walking steadily, the landing point swings back and forth in the direction of the y-axis, yet too small swing amplitude may cause interference between the legs. Hence, in the y-axis, we set the distance between the two landing points is greater than a threshold.

$$\begin{aligned} & y_{f_i} - y_{f_{i-1}} \ge \Delta_{\text{min}y}\, i = 1,2,3 \\ \text{or} \quad & y_{f_i} - y_{f_{i-1}} \ge \Delta_{\text{min}y}\, i = 1,2,3 \end{aligned} \tag{24}$$

where $\Delta_{\text{min}y}$ is the minimum step distance of the robot in the y-axis. In the next few steps, the landing point y_{f_i} should be determined according to the size of initial conditions y_{f_0} and y_{s_0}. No crossing of feet allowed.

- Step duration constraints

In the case of large disturbance, humanoid robot is limited by mechanical structure. The maximum distance of step is equal to Δ_{dx}, Δ_{dy} under the condition of fixed step timing. If it is not enough to overcome disturbance, it will lead to body forward and CoM divergence. If step duration is changeable, the robot can choose the maximum step distance to land in a shorter step time to quickly enter the next step. Through several rapid steps, it has more advantages in dealing with high-speed and large disturbances, which is similar to human beings. So when the initial velocity is obtained, the distance of one step should not exceed Δ_{dx}, Δ_{dy}

$$\begin{aligned} 0 \le x_0(2)T_{x\text{step}} \le \Delta_{dx} \\ 0 \le y_0(2)T_{y\text{step}} \le \Delta_{dy} \end{aligned} \tag{25}$$

The above method finds the real-time optimal trajectory by rolling the QP within the constrained framework, in NT time and minimizes CoM position error, velocity error, acceleration error, landing point error, and step timing error. In addition, the variation of CoP could also be minimized. Therefore, the robot can change its speed and walk with variable step size, and automatically adjust the step size and time according to the required speed or disturbance. As weight coefficients of the objective function, the parameters $[a_1, a_2, a_3, a_4, a_5, a_6]$ determines the constraint

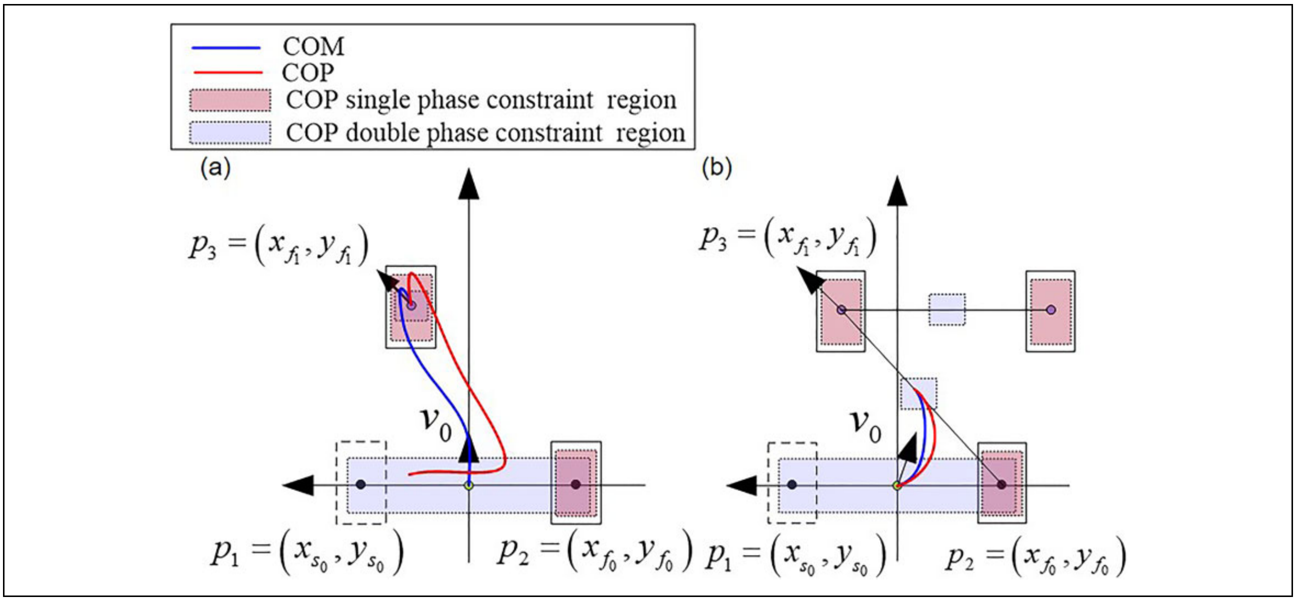
COM
COP
COP single phase constraint region
COP double phase constraint region
(a)
(b)
$p_3 = (x_{f_1}, y_{f_1})$
v_0
$p_1 = (x_{s_0}, y_{s_0})$
$p_2 = (x_{f_0}, y_{f_0})$
$p_3 = (x_{f_1}, y_{f_1})$
v_0
$p_1 = (x_{s_0}, y_{s_0})$
$p_2 = (x_{f_0}, y_{f_0})$

Figure 7. Push recovery planning process. (a) One-step stabilization mode and (b) multiple step stabilization mode.

strength of the corresponding physical quantity. The larger the coefficient, the smaller the error between the trajectory output and the target. While if the robot is disturbed, different initial state x_0 appears in each QP solution. Because of the online operation characteristics of the algorithm, the optimal output results are also changed to generate the optimal target trajectory under the disturbance.

Push recovery planning

In the same way, if the cost function and all kinds of constraints can be found manually, the rolling optimization method can be used to carry out the real-time gait planning. For push recovery, the large disturbance would grant the robot an initial velocity, and thus, the robot needs to be stabilized by step, which can be stabilized at one step or several times.

Adopt the following form of objective function

$$\begin{aligned} J = \; & \frac{a_1}{2} \| \boldsymbol{C}_x^{\text{target}} - \boldsymbol{C}_x \|^2 + \frac{a_2}{2} \| \dot{\boldsymbol{C}}_x \|^2 + \frac{a_3}{2} \| \bar{\boldsymbol{u}}_{\boldsymbol{x}} \|^2 \\ & + \frac{a_4}{2} \| \boldsymbol{x}_f^{\text{target}} - \boldsymbol{x}_f \|^2 \\ & + \frac{a_1}{2} \| \boldsymbol{C}_y^{\text{target}} - \boldsymbol{C}_y \|^2 + \frac{a_2}{2} \| \dot{\boldsymbol{C}}_{\text{y}} \|^2 + \frac{a_3}{2} \| \bar{\boldsymbol{u}}_{\text{y}} \|^2 \\ & + \frac{a_4}{2} \| \boldsymbol{y}_f^{\text{target}} - \boldsymbol{y}_f \|^2 \end{aligned} \tag{26}$$

The variable is in line with the previous section

$$\begin{cases} \boldsymbol{C}_{\boldsymbol{x}}^{\text{target}} = x_{f_1} \\ \boldsymbol{C}_y^{\text{target}} = y_{f_1} \end{cases}, \begin{cases} \boldsymbol{x}_f^{\text{target}} = x_{f_0} \\ \boldsymbol{y}_f^{\text{target}} = y_{f_0} + 0.2\text{sign}(y_{s_0} - y_{f_0)} \end{cases}$$

(as shown in Figure 7(a)), or

$$\begin{cases} \boldsymbol{C}_x^{\text{target}} = \dfrac{x_{f_0} + x_{f_1}}{2} \\ \boldsymbol{C}_y^{\text{target}} = \dfrac{y_{f_0} + y_{f_1}}{2} \end{cases}, \begin{cases} \boldsymbol{x}_f^{\text{target}} = x_{f_0} \\ \boldsymbol{y}_f^{\text{target}} = y_{f_0} + 0.2\text{sign}(y_{s_0} - y_{f_0)} \end{cases}$$

In the case of one-step stability, the robot reaches a stable state through one foot support, and the double support constraints can be selected in the form of Figure 7(a). While for multiple stride, the CoP constraint given by equation (22) in the double support, as shown in Figure 7(b), could be selected, and the optimization method of push recovery is set manually. Other structural constraints are identical to those in the "Walking planning with adjustable step duration" section.

Results and discussions

In this section, we present simulation results using the proposed walking pattern generation method. In the first scenario, the results show that the model can recover from the larger pushes when the step time adaptive controller is used. In the second scenario, we compare the maximize anti-disturbance ability of the step duration adjustable method with that of the fixed step duration method in Diedam et al.,[2] Kajita et al.,[4] and Herdt et al.[13,25]

Walking planning simulation

In this scenario, we compare our controller with the one that uses fixed step durations. Walking simulation was carried out with Matlab (version R2017b) on a personal

Table 2. System parameters.

Parameter	Value
M	3
T (s)	0.01
$\omega^2 = g/h$	9.81
$t_{x\text{step}}$ (s)	0.5
$\Delta_{dx} = \Delta_{dy}$ (m)	0.3
$[a_1, a_2, a_3, a_4, a_5, a_6]$	$[10^{-2}, 10^{-2}, 0.3, 10^{-3}, 10^{-2}, 10^{-4}]$
$[\Delta_1, \Delta_2, \Delta_3, \Delta_4]$ (m)	[0.06, 0.06, 0.04, 0.04]
$[\Delta_{x'}, \Delta_{y'}]$ (m)	[0.4, 0.2]
$[\Delta_x, \Delta_y]$ (m)	[0.06, 0.04]
Δ_{miny} (m)	0.2

computer with x64 Win10 platform (Intel(R) Core(TM) i5-7200U CPU, 8G RAM). At the beginning, the robot is in a static state in the flat ground. Using LIP model, the height of CoM remains unchanged. The state space equation (3) is used, the structural parameters of the robot (see Figure 3) are shown in Table 2. The total weight of the robot is 25.5 kg and it has 12 degrees of freedom. Each leg has three degrees of freedom (pitch, roll, yam) on the hip joint, one pitch on the knee, the ankle joint has pitch and roll degrees of freedom. Each joint is equipped with an angular displacement and torque sensors to achieve position or force closed-loop. Inertial measurement unit (IMU) units are installed on floating base coordinates.

Figures 8 to 11 show the simulation results of fixed step timing and adaptive step timing of the humanoid under disturbance. In the dual-support phase, the CoP constraint selection equation (20). $x_0 = [0, 0.2, 0, 0, 0, 0, 0, 0, 0.10, -0.10]$, $v_{x\text{ref}} = 0.5$ m/s, $v_{y\text{ref}} = 0.1$ m/s. At the initial time of 5th step, the x-direction is added a disturbance of 0.3 m/s, the y-direction is added a disturbance of 0.1 m/s. For the case when step timing is adjustable, as shown in Figures 8 and 10, the system shortens the step timing, has a recovery motion that starts with a large recovery step, and then converges to the reference motion in 1 or 2 steps. However, for the case when the step timing cannot be adjusted, as shown in Figures 9 and 11, because of the excessive disturbance, the humanoid robots uses the maximum step distance at each step, but it is still not enough to suppress the CoM divergence, which eventually leads to the robot dumping.

Comparisons with the existing methods

In the second scenario, we compare the robustness of the proposed approach with that of Herdt et al.[25] The proposed walking pattern generation approach by Herdt et al.[25] and variations[2,4,13] of this approach are standard walking pattern generators. Herdt et al.[25] calculates CoM trajectory and real-time landing position in a fixed prediction horizon and realizes the travel of predetermined speed. Here, we apply the same parameters to both methods and calculate the maximum perturbation velocities that each method can recover from different directions (see Figure 12). The measured data are based on the system parameters in the previous section. The robot has a ground contact as shown in Figure 12(a) at the beginning of the disturbance action of the 5th step, when the humanoid is in double support. At the beginning of the walking, both methods set the step timing to 0.5 s, $v_{x\text{ref}} = 0.4$ m/s, and the disturbance in the forward field of vision $(0, 180^\circ)$ was applied when the humanoid reached the 5th step. That is to say, the forward is 90° direction of forward vision (positive direction of the x-axis), the 0° of forward vision is negative direction of the y-axis, and 180° is positive direction of the y-axis.

As can be observed in Figure 12, our approach with time adjustment is able to recover from much more severe pushes compared with Herdt et al.[25] When the step time is fixed, because the robot is in double support, the resistance to disturbance in the vertical direction of the two-foot line is the weakest, and that in the direction of the two-foot line is the strongest, as shown in the green line in the Figure 12(b). With the method proposed in this article, the step timing can be adjusted, and the speed increases when the x-direction is disturbed. Under the restriction of equation (25), the step duration decreases, and the robot moves frequently, which enhances the anti-disturbance ability in the x-direction. However, in the y-direction, the minimum distance between the feet is within the range from 0.2 m to 0.3 m, because cross feet is not allowed. Even if the step time is reduced to a very small amount, it still can not produce a large CoP movement in space, as shown in Figure 13. So in the y-axis direction, the feasible area is more limited than the other direction.

Figure 12(b) compares the advantages of this method presented in this article in terms of spatial. Figure 12(c) shows the advantages of this method in terms of time. In the forward 0–180° field of view, the common recoverable disturbance velocity is as shown in the shaded area in Figure 12(b). The recovery time corresponding to the two methods is shown in Figure 12(c). It can be seen that the recovery time used in this method is less than Herdt et al.[25] when the two methods executed in the same disturbance speed in all directions. This is because the method reduces the stride time after the disturbance, but the number of steps used for recovery does not change. This in turn leads to a reduction in the recovery time used. Overall, the variable step duration method has an improved robustness as compared with the fixed step timing method.

Push recovery simulation

For push recovery, similar to human, the robot could take a big step to restore stability, or step by step to stability. As shown in Figure 14, the CoM has a initial velocity of 0.5 m/s in the x-direction and a initial velocity of 0.2 m/s in the y-direction, $\Delta_{dx} = 0.5$, $\Delta_{dy} = 0.3$. By utilizing the scheme shown in Figure 7(a) under the disturbance, it can be seen that the CoM and CoP position appear overshoot, then

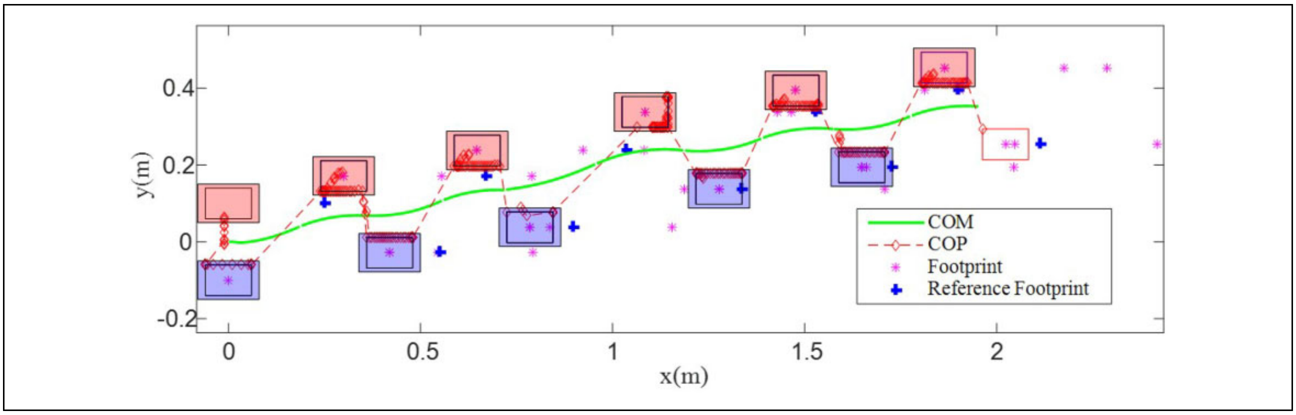

Figure 8. The CoM and CoP trajectories in the presence of perturbations under variable step timing. CoM: Center of Mass; CoP: Center of Pressure.

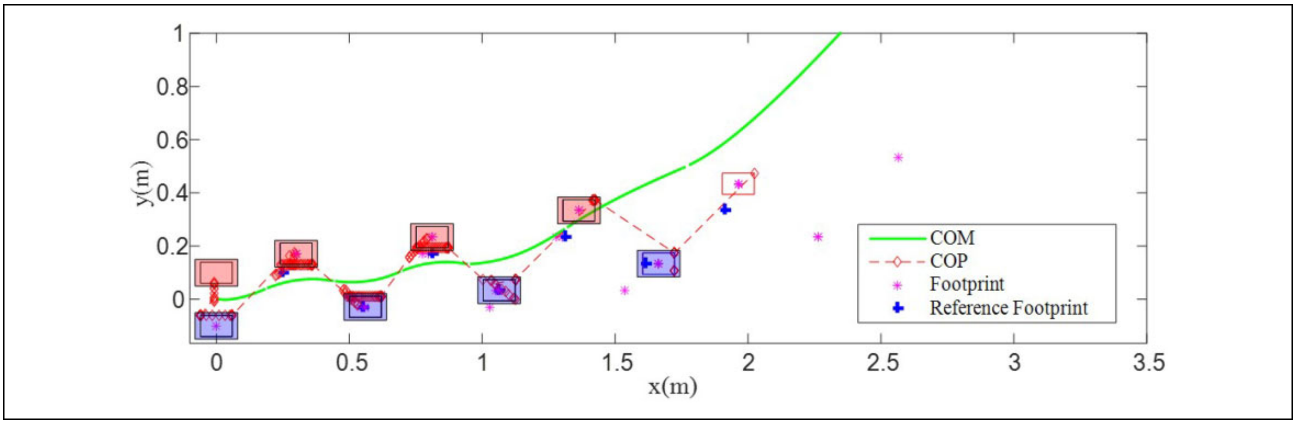

Figure 9. The CoM and CoP trajectories in the presence of perturbations under fixed step timing. CoM: Center of Mass; CoP: Center of Pressure.

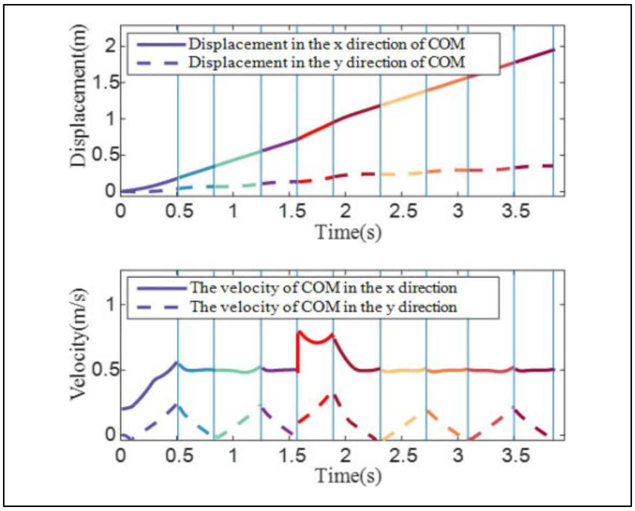

Figure 10. Variable step timing, the position and speed of CoM. CoM: Center of Mass.

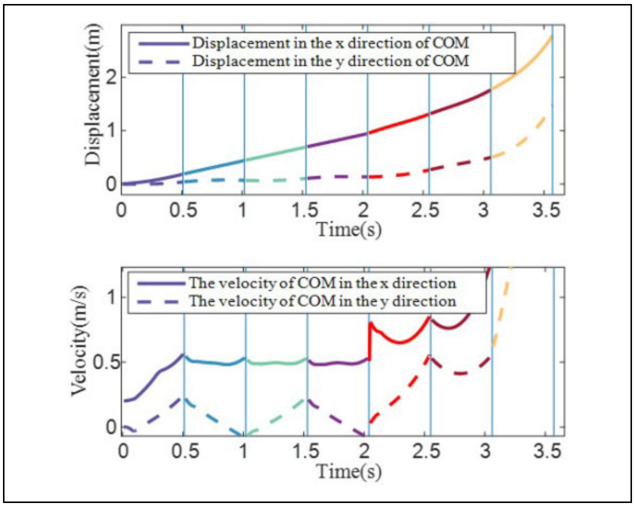

Figure 11. Fixed step timing, the position and speed of CoM. CoM: Center of Mass.

gradually tend to stabilize. At the same initial condition, using the scheme shown by Figure 7(b), the CoM gradually stabilizes after many steps, as shown in Figure 15, without overshoot. Hence, it could be known that multi-step recovery is helpful to overcome the large disturbance and is more robust than one-step method.

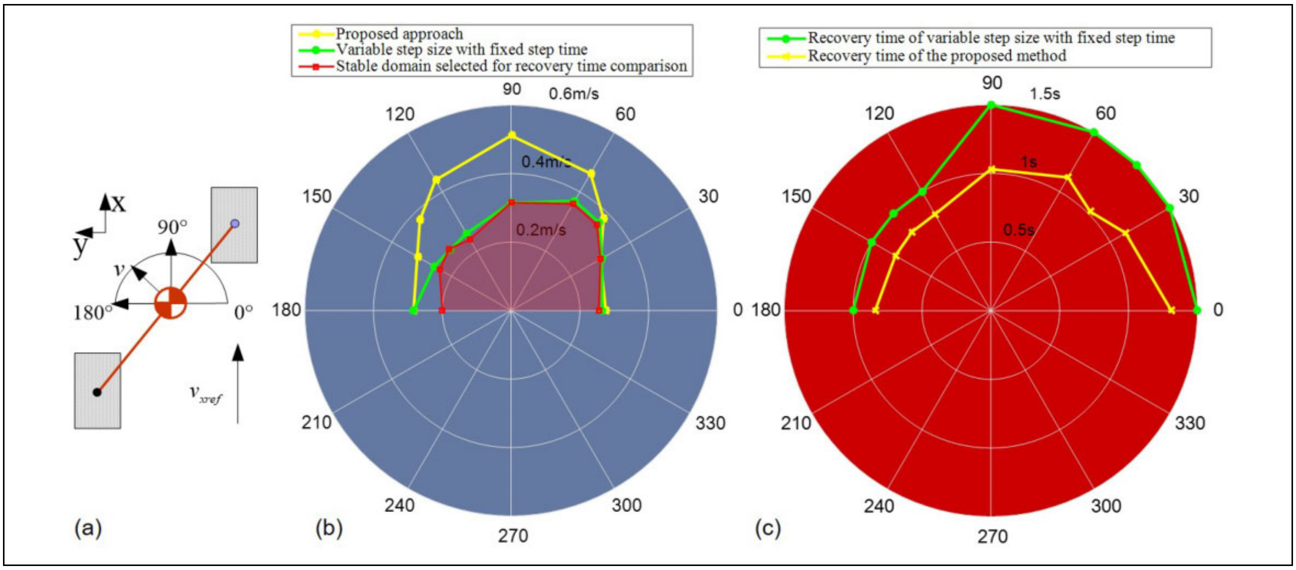

Figure 12. Contrast of maximum restorable disturbance velocities and recovery time in the front horizon. (a) Definition of frontal horizon, (b) maximum recoverable speed in the forward horizon, and (c) recovery time in the forward horizon.

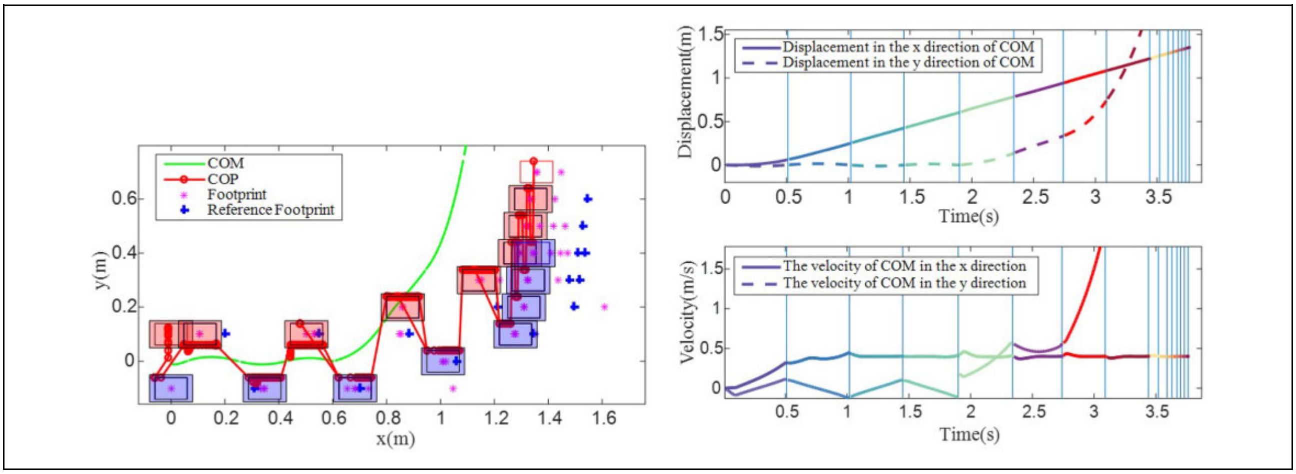

Figure 13. The instability of the humanoid lateral plane. (a) Definition of frontal horizon, (b) maximum recoverable speed in the forward horizon, and (c) recovery time in the forward horizon.

Discussion

Through the above simulation and comparative analysis, the following advantages could be achieved with the proposed method, as compared to the other existing methods,

1. *Flexibility*: The proposed method provides a variety of schemes to deal with the QP problem in the dynamic balance for humanoid robots. For the same task, there are different optimization models. While for different tasks, only minor modifications are needed to use similar optimization models with strong flexibilities.
2. *Robustness*: The simulation results show that the variable step duration method is more robust than the MPC approach with several preview steps without timing adjustment. In our method, a nominal step duration is set to allow the next prediction horizon to deviate from it in a linear constraint after disturbances, and then solve a convex optimization problem by looking at the next step location and timing.
3. *Similarity with people*: The gait generated by the method, in the single support phase, the CoP moves stably forward in the forward direction of the foot, which is quite similar to human walking. A similar approach to human is also used for the perturbation process, which results in larger CoP movements in a shorter period of time, and suppresses CoM divergence, while minimizes the speed and acceleration tracking errors.

It is worth noting that, since the main focus of this article is to test the feasibility of the proposed MPC algorithm onto

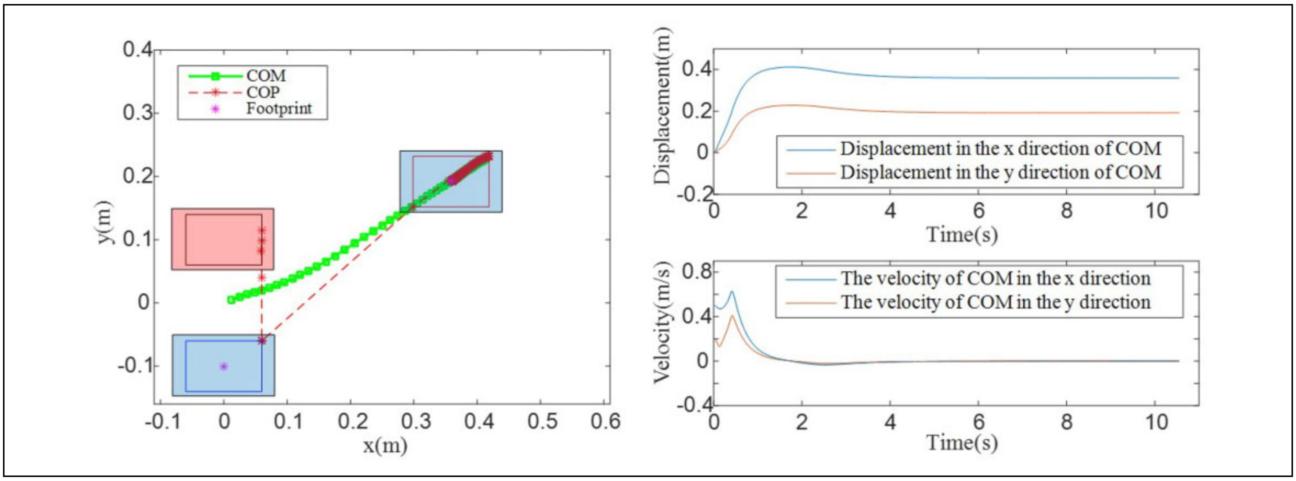

Figure 14. One-step stabilization mode for push recovery.

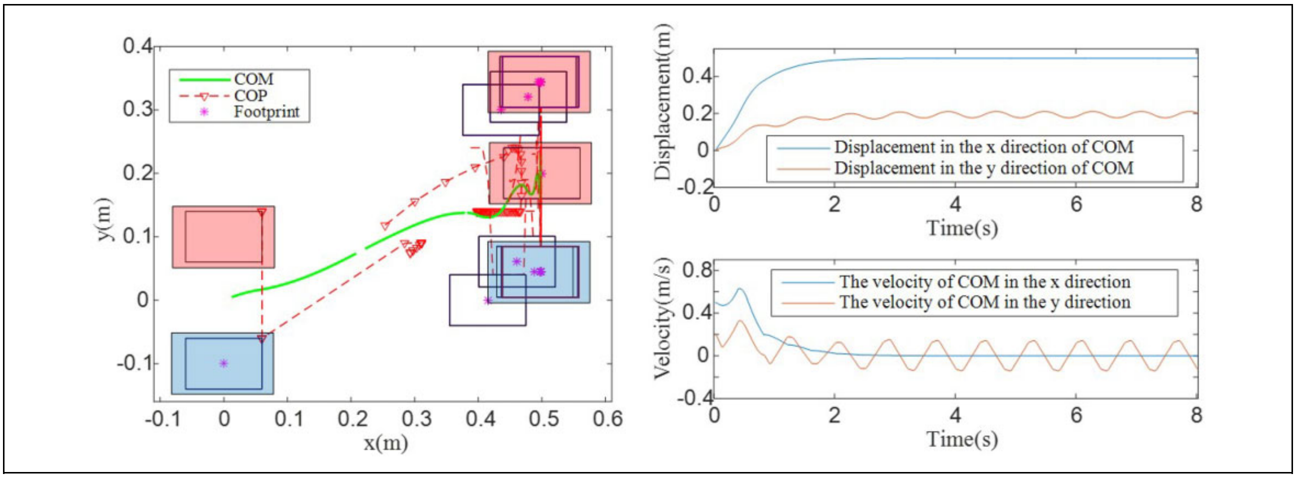

Figure 15. Multi-steps stabilization mode for push recovery.

the robot, we verified its robustness on flat ground only. When considering some other much more complicated forms of disturbances, like the uneven plane or walking on the slope, however, there may exist various other factors influencing on the performances of the robot. Those factors could impose a huge burden on the performance analysis of biped robots. Therefore, one aspect of our next-step work is to explore and improve the online anti-disturbance performances of the proposed algorithm under complex terrain.

Conclusion

In summary, we proposed an MPC method, which uses LIP as the motion model and the change of CoP as the input to minimize the variation of the CoP in the objective function, and thus guarantees the stable CoM and CoP trajectories, realizes the speed change and step duration change under the action of disturbance for humanoid robots. Compared with the fixed step duration method, the variable step distance and step timing method could further improve the compensation ability of the perturbations. Meanwhile, the flexible task of the robot is realized owing to the diversity of constraint settings. Moreover, the objective function can choose a different form by modifying some of the constraints that match it.

This study can be regarded as preliminary work for biped robots to enter human life. Our next-step work focuses mainly on the following aspects. First, we are to verify such proposed method on our lab-customized humanoid robot as shown in Figure 3, and then we would like to further extend our method to a three-dimensional space by considering the optimization in z-axis direction. Last but not least, the method could also be extended to uneven ground, within the slope or step environment to mimic the real human living environment, to further improve its adaptability.

Acknowledgments

The authors would like to acknowledge the financial support provided by China Postdoctoral Science Foundation (grant no. 2018M641013), the Natural Science Basic Research Plan in Shaanxi Province of China (program no. 2018JQ6014), and the

Fundamental Research Funds for the Central Universities (grant no. G2018KY0308).

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was financially supported by China Postdoctoral Science Foundation (grant no. 2018M641013), the Natural Science Basic Research Plan in Shaanxi Province of China (program no. 2018JQ6014), and the Fundamental Research Funds for the Central Universities (grant no. G2018KY0308).

ORCID iD

Sheng Dong https://orcid.org/0000-0001-9302-4893

References

1. Pajon A, Caron S, De Magistri G, et al. Walking on gravel with soft soles using linear inverted pendulum tracking and reaction force distribution. In: *2017 IEEE-RAS 17th international conference on humanoid robots*, pp. 432–437.
2. Diedam H, Dimitrov D, Wieber P, et al. Online walking gait generation with adaptive foot positioning through linear model predictive control. In: *2008 IEEE/RSJ international conference on intelligent robots and systems*, Nice, France, 22–26 September 2008, pp. 1121–1126.
3. Kajita S, Kanehiro F, Kaneko K, et al. The 3D linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. In: *2001 IEEE/RSJ international conference on intelligent robots and systems*, San Diego, California, USA, 29 October–2 November 2007, pp. 239–246.
4. Kajita S, Kanehiro F, Kaneko K, et al. Biped walking pattern generation by using preview control of zero-moment point. In: *2003 IEEE international conference on robotics and automation*, Taipei, Taiwan, 14–19 September 2003, pp. 1620–1626.
5. Wieber P.Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. In: *2006 IEEE-RAS 6th international conference on humanoid robots*, Genova, Italy, 4–6 December 2006, pp. 137–142.
6. Englsberger J, Ott C, Roa MA, et al. Bipedal walking control based on capture point dynamics. In: *2011 IEEE/RSJ international conference on intelligent robots and systems*, pp. 4420–4427.
7. Takenaka T, Matsumoto T, and Yoshiike T. Real time motion generation and control for biped robot—1st report: walking gait pattern generation. In: *2009 IEEE/RSJ international conference on intelligent robots and systems*, St. Louis, MO, USA, 11–15 October 2009, pp. 1084–1091.
8. Faraji S, Pouya S, Atkeson CG, et al. Versatile and robust 3D walking with a simulated humanoid robot (Atlas): a model predictive control approach. In: *2014 IEEE international conference on robotics and automation*, Hong Kong, China, 31 May–7 June 2014, pp. 1943–1950.
9. Feng S, Whitman E, Xinjilefu X, et al. Optimization-based full body control for the DARPA robotics challenge. *J Field Robot* 2014; 32: 293–312.
10. Kim S, Hirota K, Nozaki T, et al. Human motion analysis and its application to walking stabilization with COG and ZMP. *IEEE Trans Ind Info* 2018; 14(11): 5178–5186.
11. Kim S, Han Y, and Hong Y. Stability control for dynamic walking of bipedal robot with real-time capture point trajectory optimization. *J Intell Robot Syst* 2019; 1: 1–17.
12. Englsberger J, Ott C, and Albu-Schäffer A. Three-dimensional bipedal walking control based on divergent component of motion. *IEEE Trans Robot* 2015; 31(2): 355–368.
13. Herdt A, Perrin N, and Wieber P. Walking without thinking about it. In: *2010 IEEE/RSJ international conference on intelligent robots and systems*, Taipei, Taiwan, 18–22 October 2010, pp. 190–195.
14. Shen C, Zhang L, and Liu W. A stabilized filter SQP algorithm for nonlinear programming. *J Global Optim* 2016; 65(4): 677–708.
15. Fathi-Hafshejani S, Mansouri H, and Peyghami M. A large-update primal-dual interior-point algorithm for second-order cone optimization based on a new proximity function. *Optimization* 2016; 65(7): 1477–1496.
16. Kamandi A, Amini K, and Ahookhosh M. An improved adaptive trust-region algorithm. *Optim Lett* 2017; 11(3): 555–569.
17. Sun Z, Tian Y, Li H, et al. A superlinear convergence feasible sequential quadratic programming algorithm for bipedal dynamic walking robot via discrete mechanics and optimal control. *Optim Control Appl Method* 2016; 37(6): 1139–1161.
18. Kryczka P, Kormushev P, Tsagarakis NG, et al. Online regeneration of bipedal walking gait pattern optimizing footstep placement and timing. In: *2015 IEEE/RSJ international conference on intelligent robots and systems*, Hamburg, Germany, 28 September–2 October 2015, pp. 3352–3357.
19. Aftab Z, Robert T, and Wieber P.Ankle, hip and stepping strategies for humanoid balance recovery with a single model predictive control scheme. In: *2012 IEEE-RAS 12th international conference on humanoid robots*, Osaka, Japan, 29 November–1 December 2012, pp. 159–164.
20. Khadiv M, Herzog A, Moosavian SAA, et al. Step timing adjustment: a step toward generating robust gaits. In: *2016 IEEE-RAS 16th international conference on humanoid robots*, Cancun, Mexico, 15–17 November 2016, pp. 35–42.
21. Sun Z, Tian Y, and Wang J. A novel projected Fletcher-Reeves conjugate gradient approach for finite-time optimal robust controller of linear constraints optimization problem: application to bipedal walking robots. *Optim Control Appl Method* 2018; 39(1): 130–159.
22. Sun Z, Sun Y, Li Y, et al. A new trust region-sequential quadratic programming approach for nonlinear systems based on nonlinear model predictive control. *Eng Optim* 2019; 51(6): 1071–1096.

23. Regis R. Trust regions in Kriging-based optimization with expected improvement. *Eng Optim* 2016; 48(6): 1037–1059.
24. Fan S, Fan C, and Huang C. A trust region-based approach to optimize triple response systems. *Eng Optim* 2014; 46(5): 606–627.
25. Herdt A, Diedam H, Wieber PB, et al. Online walking motion generation with automatic foot step placement. *Adv Robot* 2010; 24: 719–737.
26. Kajita S, Morisawa M, Miura K, et al. Biped walking stabilization based on linear inverted pendulum tracking. In *2010 IEEE/RSJ international conference on intelligent robots and systems*, Taipei, Taiwan, 18–22 October 2010, pp. 4489–4496.
27. Tedrake R, Kuindersma S, Deits R, et al. A closed-form solution for real-time ZMP gait generation and feedback stabilization. In: *2015 IEEE-RAS 15th international conference on humanoid robots*, Seoul, South Korea, 3–5 November 2015, pp. 936–940.
28. Lanari L and Hutchinson S. Inversion-based gait generation for humanoid robots. In: *2015 IEEE/RSJ international conference on intelligent robots and systems*, Hamburg, Germany, 28 September–2 October 2015, pp. 1592–1598.
29. Scianca N, Cognetti M, De Simone D, et al. Intrinsically stable MPC for humanoid gait generation. In: *2016 IEEE-RAS 16th international conference on humanoid robots*, Cancun, Mexico, 15–17 November 2016, pp. 601–606.
30. Englsberger J, Koolen T, Bertrand S, et al. Trajectory generation for continuous leg forces during double support and heel-to-toe shift based on divergent component of motion. In: *2014 IEEE/RSJ international conference on intelligent robots and systems*, Chicago, IL, USA, 14–18 September 2014, pp. 4022–4029.
31. Ho CH, Basdogan C, and Srinivasan MA. *Numerical optimization*. Amsterdam: Elsevier, 2006.