2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

On-demand Optimal Gait Generation for a Compass Biped Robot Based on the Double Generating Function Method*

Zhiwei Hao[1], Kenji Fujimoto[2], and Yoshikazu Hayakawa[3]

Abstract— **Recently, the double generating function method for finite time linear quadratic optimal control problems was proposed. This paper applies it to the on-demand optimal gaits generation of a compass biped robot walking on the level ground. The double generating function method is employed to generate reference optimal gaits and inputs considering the energy consumption by linearizing the compass biped robot. The simulation result shows that the modeling error caused by the linearization is small when the robot walks with a reasonable step length and a appropriate time period. This implies that the optimal states and inputs for the linearized system can be treated as the optimal ones for the original nonlinear system. The biggest advantage of the double generating function method is that it can generate a parametrization of optimal gaits for different boundary conditions and different time periods. Therefore, it is very useful to generate the optimal states and inputs on demand and in real time for the real biped robots.**

I. INTRODUCTION

In the last decade, the biped locomotion problem is well studied. There already exist a lot of theoretical and experimental results of the gait generation methods for biped robots. Most of gait generation methods concerned with the stability of the biped robot use the well-known concept of Zero-Moment Point (ZMP) proposed by Vukobratovic [1]. Another important issue of concern for biped robots is energy consumption. This problem can be formalized to a standard optimal control problem which is solved by the shooting method or the parametric optimization method [2], [3]. In addition, the passive dynamic walking was proposed by McGeer more than twenty years ago from the view of saving energy [4]. In the following years, the motion analysis of the passive walking were studied by many researchers, e.g., [5], [6], [7], [8]. Although the passive walking does not need any input, it can only walk along a slight slope. Therefore, some locomotion control methods considering the energy consumption are proposed to make a biped robot walking on a level ground based on the passive walker, e.g., [9], [10], [11], [12]. Considering the case of the biped robot whose model parameters are not known exactly, an optimal gait generation method is developed using iterative learning control [13], [14].

*The first author is supported by China Scholarship Council (CSC).

[1]Z. Hao is with Department of Mechanical Science and Engineering, Nagoya University, Furo-cho, Chikusa-ku, Nagoya, Aichi 464-8603, Japan hao@haya.nuem.nagoya-u.ac.jp

[2]K. Fujimoto is with the Department of Astronautics and Aeronautics, Kyoto University, Kyotodaigaku-katsura, Nishikyo-ku, Kyoto 615-8540, JAPAN k.fujimoto@ieee.org

[3]Y. Hayakawa is with Department of Mechanical Science and Engineering, Nagoya University, Furo-cho, Chikusa-ku, Nagoya, Aichi 464-8603, Japan hayakawa@nuem.nagoya-u.ac.jp

Given a cost function which takes the energy consumption in account, the optimal trajectory generation problem has been widely studied in very diverse fields. In robotics, the reference trajectories of biped walking robots are often chosen to reduce the energy consumption (the cost function) [15]. On the other hand, when a robot is walking in a complex environment, it should be able to avoid obstacles. This implies that the initial position and/or velocity, the designed terminal position and/or velocity, and the walking time period for each step are very often different. From this view point, the optimal gait generation problem for a biped robot is equivalent to a family of optimal control problems parameterized by boundary conditions. The finite time optimal control problem can be reduced to computing the state trajectory of a Hamiltonian system. In particular, it is reduced to a two-point boundary-value problem (TPBVP) for ordinary differential equations (ODEs) [16]. There are a lot of methods, e.g. the shooting method [17] and its extension [18], to solve it. The basic principle of the shooting method is computing the trajectory repeatedly so that the exact one satisfying the boundary values is obtained. Therefore, we need to solve the TPBVP again if we change the boundary values when we use conventional methods. This will cause a heavy on-demand computation effort for the real robots.

Recently, trajectory generation methods based on the generating functions are proposed [19], [20]. Since a generating function gives a family of the optimal inputs by the canonical transformation for different boundary conditions, a family of the optimal trajectories can be obtained by numerical integration along the system dynamic equation. A recursive algorithm based on the result of [19] is proposed to solve H-JEs for the generating functions for a nonlinear system [21]. For finite-time linear quadratic optimal control problems, the authors proposed a method named the double generating function method to compute a parametrization of optimal trajectories by using a pair of generating functions [22].

This paper applies the double generating function method [22] to a compass gait biped robot on the level ground. The double generating function method gives the optimal state and input trajectories as functions of the boundary conditions, the initial time, and the terminal time directly. Since the generating functions can be calculated in advance for a given finite time linear quadratic optimal control problem, the on-demand computation time of adjusting the step length and the time period is almost none. Here, the on-demand computation time is the time calculating optimal trajectories after determining the boundary conditions in this paper. The dynamic equation of a compass biped walking robot is

978-1-4673-6357-0/13/$31.00 ©2013 IEEE

approximated by the Jacobian linearization to us the double generating function method. The optimal state and input trajectory generated for the linear model is taken as the reference one. A conventional PD controller is designed to track the reference state trajectory with the reference input for the original nonlinear model. Therefore, the modeling error by the linearization can be shown by comparing the state and input trajectories generated by the simulation of the nonlinear control system with the reference ones. The simulation result shows the modeling error caused by the linearization is very small when the robot walks in a low speed with a reasonable size of the step length. At the same time, both of them are not so large. This means that the optimal state and input trajectories for the linearized system can be used as the optimal ones for the original nonlinear system in practice. Because of the advantage of the double generating function method, it would be very useful for controlling the real biped robots, especially, when they are walking in a complex environment.

This paper is organized as follows. The preliminaries of the finite time linear quadratic optimal control problem and the double generating function method are introduced in Section 2. Section 3 addresses the optimal gaits generation problem for the biped robot and designs a PD controller to analyze the modeling error caused by the linear approximation. The simulation and the error analysis illustrate the effectiveness and the usefulness of the double generating function method for controlling the real biped robots in Section 4.Section 5 concludes the paper.

II. PRELIMINARIES

This section introduces some preliminaries of finite time linear optimal control problems and how to solve it using the double generating function method proposed in [22].

A. Linear Optimal Control Problem

A finite time linear optimal control problem and how it is rendered to a two-point boundary-value problem (TPBVP) for ordinary differential equations (ODEs) are introduced in this subsection [23]. Given a linear finite time optimal control problem, the system equation is

$$\dot{x} = Ax + Bu, \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the state, $A \in \mathbb{R}^{n\times n}$ is a constant matrix, $B \in \mathbb{R}^{n\times m}$ is a constant matrix, $u \in \mathbb{R}^m$ is the input. Define a cost function

$$J(x_0, u) = \frac{1}{2}\int_{t_0}^{t_f} (x^{\mathrm{T}}Qx + u^{\mathrm{T}}Ru)\mathrm{d}t, \tag{2}$$

where the symbols $x_0 \in \mathbb{R}^n$ is the initial value of the state, t_0 and t_f are initial and terminal time respectively, $Q \in \mathbb{R}^{n\times n}$ is a semi-positive definite matrix, and $R \in \mathbb{R}^{m\times m}$ is a positive definite matrix. The purpose is to find an optimal input $u^*(t)$ minimizing the cost function J as

$$u^*(t) = \arg\min_u J(x_0, u) \tag{3}$$

subject to the boundary condition

$$x(t_0) = x_0, \quad x(t_f) = x_{t_f}, \tag{4}$$

where $x_{t_f} \in \mathbb{R}^n$ is the terminal value of the state.

Let us introduce a column vector $\lambda \in \mathbb{R}^n$ to represent the costate, according to Pontryagin's minimum principle [23], a necessary condition for the minimum of the performance index in (2) is

$$\dot{x} = H_\lambda(x,\lambda,u), \quad \dot{\lambda} = -H_x(x,\lambda,u), \tag{5}$$

$$u^* = -R^{-1}B^{\mathrm{T}}\lambda, \tag{6}$$

where the boundary value is given in (4), $H_{(\cdot)}$ denotes the partial derivative $\partial H/\partial(\cdot)$, and $H(\cdot)$ is the Hamiltonian function which is defined as

$$H(x,\lambda) = \frac{1}{2}x^{\mathrm{T}}Qx + \lambda^{\mathrm{T}}Ax - \frac{1}{2}\lambda^{\mathrm{T}}BR^{-1}B^{\mathrm{T}}\lambda. \tag{7}$$

Therefore, the optimal control problem (1)-(4) is equivalent to the TPBVP for ODEs (5). We will use the double generating function method to solve it.

B. Double Generating Function Method [22]

This subsection explains the double generating function method for a TPBVP. This method can generate a family of optimal trajectories with almost no on-demand computation for different boundary conditions. Given the linear Hamiltonian system (5), there exist four types of generating functions S_1, S_2, S_3, and S_4. Every type of the generating function has both forward and backward versions (denoted by the subscript f and b respectively). In fact, the first type and the fourth type of generating function are not well-defined at the initial time or the terminal time [19]. Among the left well-defined generating functions, any two can give the optimal trajectories and inputs. However, the initial value of the costate λ_0 and/or the terminal value of the costate λ_{t_f} are necessary when we use other pairs of generating functions except for the pair of $S_{3f}(\lambda,x_0,t)$ and $S_{3b}(\lambda,x_{t_f},t)$. Furthermore, it will cause numerical instability as the time interval increases if we use a pair of generating functions with the same time direction [22]. Therefore, the pair of $S_{3f}(\lambda,x_0,t)$ and $S_{3b}(\lambda,x_{t_f},t)$ is more convenient to generate optimal state and input trajectories for different boundary conditions than the other pairs of generating functions.

The generating functions $S_{3f}(\lambda,x_0,t)$ and $S_{3b}(\lambda,x_{t_f},t)$ have the following forms respectively:

$$S_{3f}(\lambda,x_0,t) = \frac{1}{2}\lambda^{\mathrm{T}}Z_{3f}(t)\lambda + \lambda^{\mathrm{T}}Y_{3f}(t)x_0 + \frac{1}{2}x_0^{\mathrm{T}}W_{3f}(t)x_0, \tag{8}$$

$$S_{3b}(\lambda,x_{t_f},t) = \frac{1}{2}\lambda^{\mathrm{T}}Z_{3b}(t)\lambda + \lambda^{\mathrm{T}}Y_{3b}(t)x_{t_f} + \frac{1}{2}x_{t_f}^{\mathrm{T}}W_{3b}(t)x_{t_f}. \tag{9}$$

They provide the following canonical transformation:

$$x = -\frac{\partial S_{3f}(\lambda,x_0,t)}{\partial\lambda}, \quad \lambda_0 = -\frac{\partial S_{3f}(\lambda,x_0,t)}{\partial x_0}, \tag{10}$$

$$x = -\frac{\partial S_{3b}(\lambda,x_{t_f},t)}{\partial\lambda}, \quad \lambda_{t_f} = -\frac{\partial S_{3b}(\lambda,x_{t_f},t)}{\partial x_{t_f}}. \tag{11}$$

Moreover, they satisfy the corresponding HJE respectively [19]:

$$\frac{\partial S_{3f}(\lambda, x_0, t)}{\partial t} + H(-\frac{\partial S_{3f}}{\partial \lambda}, \lambda) = 0, \quad (12)$$

$$\frac{\partial S_{3b}(\lambda, x_{t_f}, t)}{\partial t} + H(-\frac{\partial S_{3b}}{\partial \lambda}, \lambda) = 0. \quad (13)$$

Due to (10) and (11), the initial values of $S_{3f}(\lambda, x_0, t)$ and the terminal values of $S_{3b}(\lambda, x_{t_f}, t)$ are as follows respectively:

$$S_{3f}(\lambda, x_0, t_0) = -\lambda_0^{\mathrm{T}} x_0, \quad S_{3b}(\lambda, x_{t_f}, t_f) = -\lambda_{t_f}^{\mathrm{T}} x_{t_f}. \quad (14)$$

By solving the first equation in (10) and (11) for $x(t)$ and $\lambda(t)$, the following theorem is proposed, which gives a parametrization of optimal trajectories and inputs for different boundary conditions and different time intervals for a finite time linear optimal control problem.

Theorem 1. [22] Suppose that the matrices $Z_{3f}(t)$, $Y_{3f}(t)$, $Z_{3b}(t)$, and $Y_{3b}(t) \in \mathbb{R}^{n \times n}$, $0 \le t \le T$, satisfy the following ODEs

$$\dot{Z}_{3f}(t) = Z_{3f}^{\mathrm{T}}(t) A^{\mathrm{T}} + A Z_{3f}(t) - Z_{3f}^{\mathrm{T}}(t) Q Z_{3f}(t) + B R B^{\mathrm{T}}, \quad (15)$$

$$\dot{Y}_{3f}(t) = (A - Z_{3f}^{\mathrm{T}}(t) Q) Y_{3f}(t), \quad (16)$$

$$\dot{Z}_{3b}(t) = -Z_{3b}^{\mathrm{T}}(t) A^{\mathrm{T}} - A Z_{3b}(t) + Z_{3b}^{\mathrm{T}}(t) Q Z_{3b}(t) - B R B^{\mathrm{T}}, \quad (17)$$

$$\dot{Y}_{3b}(t) = (-A + Z_{3b}^{\mathrm{T}}(t) Q) Y_{3b}(t), \quad (18)$$

with the initial conditions

$$Z_{3f}(0) = 0, \quad Y_{3f}(0) = -I, \quad (19)$$

$$Z_{3b}(0) = 0, \quad Y_{3b}(0) = -I. \quad (20)$$

Then, for the finite time linear optimal control problem (1)-(4), the optimal state $x^*(t, t_0, x_0, t_f, x_{t_f})$ $(t \in [t_0, t_f])$ and input $u^*(t, t_0, x_0, t_f, x_{t_f})$ $(t \in [t_0, t_f])$, $t_f \le t_0 + T$, are given by

$$\begin{pmatrix} x^*(t, x_0, x_{t_f}, t_0, t_f) \\ u^*(t, x_0, x_{t_f}, t_0, t_f) \end{pmatrix} = \left(\begin{array}{l} -Z_{3f}(t-t_0) \cdot \\ -R^{-1} B^{\mathrm{T}} \cdot \\ \left[Z_{3b}(t_f - t) - Z_{3f}(t - t_0) \right]^{-1} \left[Y_{3f}(t-t_0) x_0 - Y_{3b}(t_f - t) x_{t_f} \right] \\ \left[Z_{3b}(t_f - t) - Z_{3f}(t - t_0) \right]^{-1} \left[Y_{3f}(t-t_0) x_0 - Y_{3b}(t_f - t) x_{t_f} \right] \\ -Y_{3f}(t-t_0) x_0 \end{array} \right). \quad (21)$$

Here $x_0 = x(t_0)$ and $x_{t_f} = x(t_f)$ are the given initial and terminal values of the state respectively.

Theorem 1 claims that once we obtain matrices $Z_{3f}(t)$, $Y_{3f}(t)$, $Z_{3b}(t)$, and $Y_{3b}(t)$ by calculating the numerical integration of (15)-(18), which are the coefficient matrices of the generating functions of $S_{3f}(\lambda, x_0, t)$ and $S_{3b}(\lambda, x_{t_f}, t)$, then we can readily obtain the optimal state and input trajectories for different boundary conditions as in (21).

III. OPTIMAL GAIT GENERATION FOR THE COMPASS BIPED WALKING ROBOT

Firstly, the model of the compass biped walking robot is introduced in the section. Then how to use double generating function method to generate the reference optimal trajectories and inputs is elaborated. At last, a PD controller is designed to analyze the modeling error caused by the linearization.

A. The Compass Biped Walking Robot

This subsection introduces a compass gait biped robot. A walking robot, named the compass gait biped [5], can walk down a gentle slope under appropriate initial conditions [4]. Now, let us consider it walks on the level ground with input, which is depicted in Fig. 1. The physical parameters and variables are shown in Table I. The main modeling assumptions are listed as follows (for the other common assumptions, refer to [5]).

Assumption 1. The transition of the supporting leg occurs instantaneously when the swinging leg touches the ground and previous supporting leg leaves the ground.

Assumption 2. The collision of the swinging leg with the ground is assumed to be inelastic and without sliding.

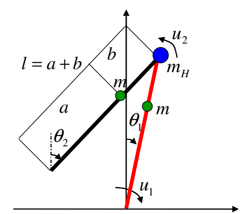


Fig. 1. Compass Gait Biped

TABLE I
PARAMETERS AND VARIABLES

Notation	Meaning	Unit
m_H	hip mass	kg
m	leg mass	kg
a	length from m to ground	m
b	length from hip to m	m
l	total leg length	m
g	gravity acceleration	m/s^2
θ_1	supporting leg angle w.r.t. vertical	rad
θ_1	swinging leg angle w.r.t. vertical	rad
u_1	ankle torque	Nm
u_2	hip torque	Nm
θ	$(\theta_1, \theta_2)^{\mathrm{T}}$	
$\dot{\theta}$	$(\dot{\theta}_1, \dot{\theta}_2)^{\mathrm{T}}$	
u	$(u_1, u_2)^{\mathrm{T}}$	

Then the compass gait biped robot with energy input in Fig. 1 can be modeled as [5]

$$M(\theta)\ddot{\theta} + N(\theta, \dot{\theta})\dot{\theta} + G(\theta) = Cu, \quad (22)$$

where

$$M(\theta) = \begin{bmatrix} m_H l^2 + m a^2 + m l^2 & -mlb\cos(\theta_1 - \theta_2) \\ -mlb\cos(\theta_1 - \theta_2) & mb^2 \end{bmatrix}, \quad (23)$$

$$N(\theta, \dot{\theta}) = \begin{bmatrix} 0 & -mlb\dot{\theta}_2 \sin(\theta_1 - \theta_2) \\ mlb\dot{\theta}_1 \sin(\theta_1 - \theta_2) & 0 \end{bmatrix}, \quad (24)$$

and

$$G(\theta) = \begin{bmatrix} -(m_H l + m(a+l)) g \sin\theta_1 \\ mgb\sin\theta_2 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}. \quad (25)$$

B. Optimal Gait Generation For The Linearized System

Defining the state x as

$$x = (\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)^{\mathrm{T}} = (x_1, x_2, x_3, x_4)^{\mathrm{T}}, \tag{26}$$

then the linearized dynamic equation of the biped robot in Fig. 1 is written as

$$\dot{x} = Ax + Bu, \tag{27}$$

where $u = (u_1, u_2)^{\mathrm{T}}$,

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{g(m_H l + ma + ml)}{m_H l^2 + ma^2} & -\frac{mgl}{m_H l^2 + ma^2} & 0 & 0 \\ \frac{gl(m_H l + ma + ml)}{b(m_H l^2 + ma^2)} & -\frac{g(m_H l^2 + ma^2 + ml^2)}{b(m_H l^2 + ma^2)} & 0 & 0 \end{bmatrix}, \tag{28}$$

and

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{m_H l^2 + ma^2} & \frac{l-b}{b(m_H l^2 + ma^2)} \\ \frac{l}{b(m_H l^2 + ma^2)} & \frac{m_H l^2 + ma^2 + ml^2 - mbl}{mb^2(m_H l^2 + ma^2)} \end{bmatrix}. \tag{29}$$

Let us consider a cost function which takes the energy consumption into account as in (2). The objective is to find a family of optimal inputs $u^*(t)$ minimizing the cost function J subject to a family of the boundary condition. In practice, both of the ankle input and the hip one should not be too large because of the limitation of the actuators. We should choose the appropriate weighting matrix R of the cost function (2) with respect to the input and the time period $t_f - t_0$. Then, a family of optimal state and input trajectories for different boundary conditions and different time intervals are given by Theorem 1. The following algorithm is obtained readily to generate the reference trajectories for the biped robots.

Algorithm 1

1. *Linearize the dynamic equation in (22) to obtain the matrices A and B as in (28) and (29) respectively;*
2. *Determine the matrices Q and R in (2);*
3. *Select T (larger than or equal to the largest designed time period for the given robot) and integrate (15)-(18) numerically forward from* 0 *to T with the initial values in (19) and (20);*
4. *Use (21) to generate the optimal state trajectory and optimal input;*
5. *If we change the boundary condition and/or terminal time* t_f *and/or initial time* t_0 ($t_f - t_0 \leq T$)*, go to step 4.*

This algorithm implies that it is very convenient to change the boundary condition and/or the time period for each step for the biped robots, since the on-demand computation for this change is almost none. However, these trajectories and inputs are optimal for the linearized system. Therefore, if we want to treat these trajectories and inputs as the optimal ones for the original nonlinear system, it is necessary to analyze the modeling error caused by the linear approximation. The following subsection will introduce a conventional PD controller to follow the reference trajectories. Then we can analyze the modeling error by comparing the state and the input trajectories of the simulation result of the original nonlinear system with the reference ones.

C. Trajectory Tracking Based on PD Control

It is well known that most of industrial manipulators are equipped with the simplest proportional and derivative (PD) controller. Now, let us consider the conventional PD controller to track the reference trajectory with the reference input for the compass biped walking robot based on the input generated by the double generating function method.

The control law is designed for the dynamic equation of compass biped walking robot (22) as follows:

$$\tau = K_\theta(\theta^{ref} - \theta) + K_{\dot{\theta}}(\dot{\theta}^{ref} - \dot{\theta}) + u^{ref}, \tag{30}$$

where, K_θ and $K_{\dot{\theta}}$ are diagonal constant matrices, they play the P-gain and the D-gain role respectively, and $\theta^{ref} = (x_1^*, x_2^*)^{\mathrm{T}} = (\theta_1, \theta_2)^{\mathrm{T}}$, $\dot{\theta}^{ref} = (x_3^*, x_4^*)^{\mathrm{T}} = (\dot{\theta}_1, \dot{\theta}_2)^{\mathrm{T}}$, and $u^{ref} = (u_1^*, u_2^*)^{\mathrm{T}}$ are the optimal trajectory and input respectively generated by the double generating function method for the linearized compass biped walking robot. Fig. 2 shows the structure of the PD controller for the compass biped walking robot.

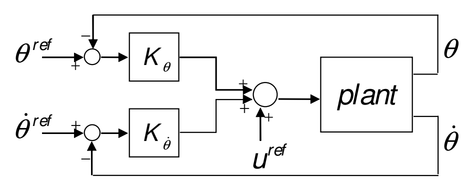


Fig. 2. The PD controller

IV. SIMULATION AND RESULT ANALYSIS

This section gives the simulation result. We take the compass gait biped robot in Fig. 1 with the parameters $a = 0.5$[m], $b = 0.5$[m], $l = 1.0$[m], $m = 5$[m], $m_H = 10$[Kg], and $g = 9.8$[m/s^2] as an example. We select the design parameters $Q = I$ (the identity matrix) and $R = \mathrm{diag}(10, 1)$ based on some heuristic simulations. If $R = \mathrm{diag}(1, 1)$, the ankle torque is larger than the hip one; if $R = \mathrm{diag}(100, 1)$, the ankle torque can be very small, but the hip torque will be very large. Due to the limitation of the space, we only show a few of simulation result. The P-gain and D-gain of the PD controller in Fig. 2 are designed as $K_\theta = \mathrm{diag}(100, 50)$, $K_{\dot{\theta}} = \mathrm{diag}(50, 20)$.

In what follows, the time period of one step for the biped robot begins from the time just after the previous transition of the supporting leg and ends at the time just before the next transition of the supporting leg. Then the initial and terminal values of the states for one step are those of the states at the time just after the previous impact and that just before the next impact respectively. The optimal trajectories and inputs generated by the double generating function method for the linearized system are taken as the reference ones. For any figure in the remainder of this paper, the dashed lines denote the reference state and input trajectories and the solid lines denote the state and the input ones generated by employing the PD controller for the original nonlinear system.

The biggest advantage of the double generating function method is that it can generate a parametrization of optimal trajectories and inputs for different boundary conditions. To show the advantage the double generating function method, we will consider the continuous walking for the biped robot in Fig. 1. Since a continuous walking contains many steps, it is necessary to introduce the transition model for different steps firstly. Assumption 2 implies that the robot configuration does not changed during the instantaneous transition stage [24], then

$$\theta^+ = \Gamma\theta^-, \tag{31}$$

with:

$$\Gamma = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{32}$$

Here, the matrix Γ exchanges the supporting and the swinging leg angles for the upcoming swinging stage. The pre-impact and post-impact variables are identified with the superscripts $-$ and $+$ respectively. Similarly, it is very easy to know that

$$\theta^- = \Gamma\theta^+. \tag{33}$$

Assumption 2 also implies that the angular momentum of the robot about the impacting foot as well as the angular momentum of the pre-impact supporting leg about the hip are conserved. These conservation laws lead to a discontinuous change in robot velocity. This reads:

$$\Phi^-(\theta^-)\dot{\theta}^- = \Phi^+(\theta^+)\dot{\theta}^+, \tag{34}$$

where,

$$\Phi^-(\theta^-) = \begin{bmatrix} -mab + (m_H l^2 + 2mal)\cos(\theta_2^- - \theta_1^-) & -mab \\ -mab & 0 \end{bmatrix}, \tag{35}$$

$$\Phi^+(\theta^+) = \begin{bmatrix} ml(l - b\cos(\theta_1^+ - \theta_2^+)) + ma^2 + m_H l^2, & mb(b - l\cos(\theta_1^+ - \theta_2^+)) \\ -mbl\cos(\theta_1^+ - \theta_2^+), & mb^2 \end{bmatrix}. \tag{36}$$

Due to (31) and (34), the post-impact angular velocity of the legs of the robot can be derived from the pre-impact angular velocity of the legs as

$$\begin{aligned} \dot{\theta}^+ &= (\Phi^+(\theta^+))^{-1}\Phi^-(\theta^-)\dot{\theta}^- \\ &= (\Phi^+(\Gamma\theta^-))^{-1}\Phi^-(\theta^-)\dot{\theta}^-. \end{aligned} \tag{37}$$

Now, let us consider a family of boundary conditions for seven steps, the compass biped robot begins walking from the stationary state and ends at the stationary state. The terminal value of the first step is determined heuristically, e.g.,

$$x(t_f) = (0.1, -0.1, 0.4, -0.4)^{\mathrm{T}}. \tag{38}$$

The time period is designed heuristically as 1.0 for the first three steps and it is 0.8 for the left four steps, so the boundary condition of the first step is

$$\begin{cases} x^1(t_0^1) = (0,0,0,0)^{\mathrm{T}}, & t_0^1 = 0, \\ x^1(t_f^1) = (0.1, -0.1, 0.4, -0.4)^{\mathrm{T}}, & t_f^1 = 1.0^-. \end{cases} \tag{39}$$

Here, $x^i(t)$, t_0^i, and t_f^i denote the state at the time t, the initial time, and the terminal time for the i-th step, respectively, and t^+ (or t^-) denotes the time just after (or just before) t. The second step and the third step are designed as a periodic gait, the step length of the fourth step is designed to be larger than the one of the previous step, e.g.,

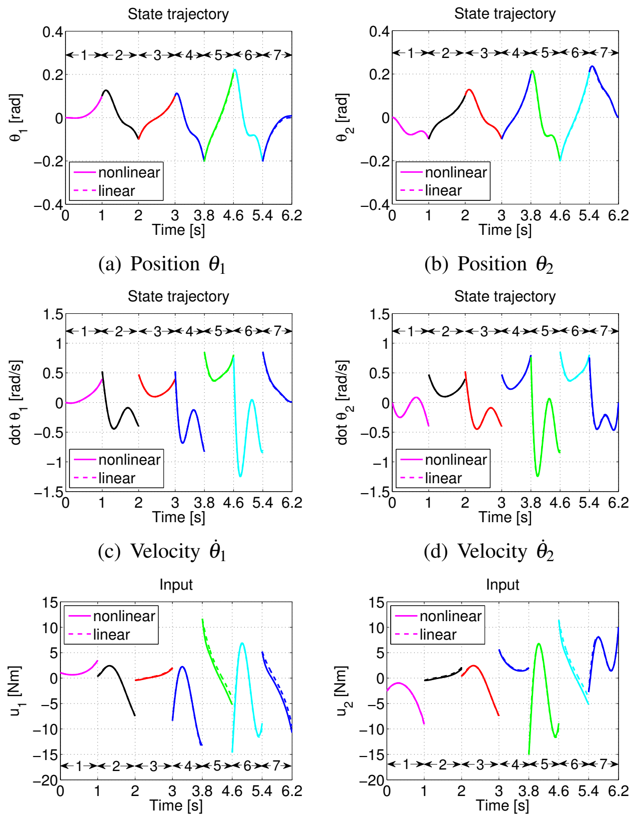
$$x^4(t_f^4) = (0.2, -0.2, 0.8, -0.8)^{\mathrm{T}}, \tag{40}$$

and the fifth and sixth steps are also designed as a periodic gait. Finally, the biped robot stops walking. According to the transition model (31) and (37), the boundary conditions of the left six steps are calculated as

$$\begin{cases} x^2(t_0^2) = (-0.1, 0.1, 0.44, 0.46)^{\mathrm{T}}, & t_0^2 = 1.0^+, \\ x^2(t_f^2) = (0.1, -0.1, 0.40, -0.40)^{\mathrm{T}}, & t_f^2 = 2.0^-, \\ x^3(t_0^3) = (-0.1, 0.1, 0.44, 0.46)^{\mathrm{T}}, & t_0^3 = 2.0^+, \\ x^3(t_f^3) = (0.1, -0.1, 0.40, -0.40)^{\mathrm{T}}, & t_f^3 = 3.0^-, \\ x^4(t_0^4) = (-0.1, 0.1, 0.44, 0.46)^{\mathrm{T}}, & t_0^4 = 3.0^+, \\ x^4(t_f^4) = (0.2, -0.2, 0.80, -0.80)^{\mathrm{T}}, & t_f^4 = 3.8^-, \\ x^5(t_0^5) = (-0.2, 0.2, 0.81, 0.69)^{\mathrm{T}}, & t_0^5 = 3.8^+, \\ x^5(t_f^5) = (0.2, -0.2, 0.80, -0.80)^{\mathrm{T}}, & t_f^5 = 4.6^-, \\ x^6(t_0^6) = (-0.2, 0.2, 0.81, 0.69)^{\mathrm{T}}, & t_0^6 = 4.6^+, \\ x^6(t_f^6) = (0.2, -0.2, 0.80, -0.80)^{\mathrm{T}}, & t_f^6 = 5.4^-, \\ x^7(t_0^7) = (-0.2, 0.2, 0.81, 0.69)^{\mathrm{T}}, & t_0^7 = 5.4^+, \\ x^7(t_f^7) = (0,0,0,0)^{\mathrm{T}}, & t_f^7 = 6.2, \end{cases} \tag{41}$$

respectively.

Fig. 3 shows the state and the input trajectories for the above designed seven steps, where the line in the i-th ($i = 1, 2, \cdots, 7$) time interval denotes the corresponding trajectory of the i-th step. Fig. 4 shows he phase portrait of the steps 2-3 and the steps 5-6, where the dashed and dotted lines denote the velocity jumps occurred when the supporting leg and the swinging leg exchange. Note that, θ_1 (θ_2) and $\dot{\theta}_1$ ($\dot{\theta}_2$) are the angle of the supporting (swinging) leg and the angular velocity, respectively. The supporting (swinging) leg may be different for different steps in fact, but θ_1 (θ_2) and $\dot{\theta}_1$ ($\dot{\theta}_2$) stick to the same leg in Figs. 3 and 4. We can see that both of the state and the input trajectories for the original nonlinear system generated by the PD controller are almost the same with the reference ones from Fig. 3. This implies that the modeling error caused by the linear approximation is very small so that we can neglect it. Therefore, the optimal gaits and inputs for the linearized model can be used as the optimal ones for the original nonlinear model. What is the important is that Fig. 3 (e)-(f) show that both of the generated ankle input and the hip input are small (not larger than 15 $[Nm]$). Therefore, they are useful in practice. In Fig. 4, it exhibits that the closed loops for steps 2-3 and 5-6 which imply the periodic motions generated.

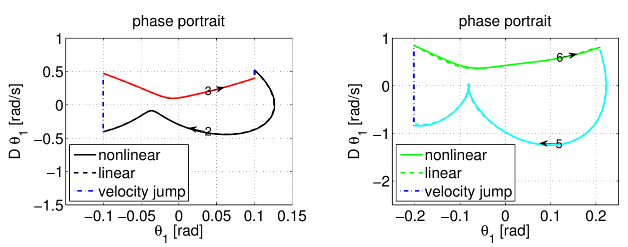


(a) Position θ_1 (b) Position θ_2

(c) Velocity $\dot{\theta}_1$ (d) Velocity $\dot{\theta}_2$

(e) Input u_1 (f) Input u_2

Fig. 3. State and input trajectories for the periodic gaits



(a) steps 2 & 3 (b) steps 5 & 6

Fig. 4. Phase portrait of $\theta_1 - \dot{\theta}_1$ for the periodic gaits

V. CONCLUSIONS

This paper shows the possibility of the application of the double generating function method to the optimal gait generation for a compass biped robot. Since the double generating function method is for the linear system, the dynamic equation of the biped robot is linearized. A PD controller is designed to analyze the modeling error caused by the linearization. The simulation result shows that the modeling error caused by the linearization is small when the robot walks with a reasonable step length and a appropriate time period. Because of the advantage of the double generating function method, it is very convenient to generate the optimal gaits and inputs on demand and in real time for the different condition and the different time periods. Therefore, it would be very useful for controlling the real robots. However, this paper does not consider the condition of contact with the ground for the biped robot, this will be studied in the future.

REFERENCES

[1] M. Vukobratovic and D. Juricic, "Contribution to the synthesis of biped gait," *IEEE Transactions on Biomedical Engineering*, vol. 16, no. 1, pp. 1–6, 1969.

[2] M. Rostami and G. Bessonnet, "Sagittal gait of a biped robot during the single support phase. part 2: Optimal motion," *Robotica*, vol. 19, pp. 241–253, 2001.

[3] T. Saidouni and G. Bessonnet, "Generating globally optimised sagittal gait cycles of a biped robot," *Robotica*, vol. 21, pp. 199–210, 2003.

[4] T. McGeer, "Passive dynamic walking," *International Journal of Robotics Research*, vol. 9, no. 2, pp. 62–82, 1990.

[5] A. Goswami, B. Thuilot, and B. Espiau, "Compass-like biped robot part i: Stability and bifurcation of passive gaits," *INRIA Research Report*, no. 2996, 1996.

[6] M. Garcia, A. Chatterjee, A. Ruina, and M. Coleman, "The simplest walking model: Stability, complexity, and scaling," *ASME J. Biomechanical Eng.*, vol. 120, pp. 281–288, 1996.

[7] K. Osuka and K. Kirihara, "Motion analysis and experimentsof passive walking robot quartet ii," in *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, 2000, pp. 3052–3056.

[8] A. Sano, Y. Ikemata, and H. Fujimoto, "Analysis of dynamics of passive walking from storage energy and supply rate," in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, 2003, pp. 2478–2483.

[9] A. Goswami, B. Thuilot, and A. Keramane, "Limit cycles in a passive compass gait biped and passivity-mimicking control laws," *Autonomous Robots*, vol. 4, no. 3, pp. 273–286, 1997.

[10] M. W. Spong, "Passivity-base control of the compass gait biped," in *Proceedings of the IFAC World Congress*, 1999, pp. 19–23.

[11] F. Asano, M. Yamakita, N. Kamamichi, and Z. W. Luo, "A novel gait generation for biped walking robots based on mechanical energy constraint," *IEEE Trans. Robotics and Automation*, vol. 20, no. 3, pp. 565–573, 2004.

[12] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science*, vol. 307, no. 5712, pp. 1082–1085, 2005.

[13] S. Satoh, K. Fujimoto, and S. Hyon, "A framework for optimal gait generation via learning optimal control using virtual constraint," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 3426–3432.

[14] K. Fujimoto and T. Sugie, "Iterative learning control of hamiltonian systems: I/o based optimal control approach," *IEEE Trans. Autom. Contr.*, vol. 48, no. 10, pp. 1756–1761, 2003.

[15] V. Duindam and S. Stramigioli, *Modeling and Control for Efficient Bipedal Walking Robots*, 1st ed. Berlin: Springer, 2009, pp. 129–161.

[16] A. Locatelli, *Optimal Control: An Introduction*, 1st ed. Basel-Boston-Berlin: Birkhauser Verlag, 2001, pp. 147–164.

[17] H. B. Keller, *Numerical Methods for Two-Point Boundary-Value Problems*. Waltham, Massachusetts: Blaisdell, 1968, pp. 39–68.

[18] E. Cristiani and P. Martinon, "Initialization of the shooting method via the hamilton-jacobi-bellman approach," *Journal of Optimization Theory and Applications*, vol. 146, no. 2, pp. 321–346, 2010.

[19] V. Guibout and D. J. Scheeres, "Solving relative two point boundary value problems: Spacecraft formation flight transfers application," *J. Guid. Control. Dyn.*, vol. 27, no. 4, pp. 693–704, 2004.

[20] C. Park and D. J. Scheeres, "Determination of optimal feedback terminal controllers for general boundary conditions using generating functions," *Automatica*, vol. 42, no. 5, pp. 869–875, 2006.

[21] Z. Hao and K. Fujimoto, "Approximate solutions to the hamilton-jacobi equations for generating functions with a quadratic cost function with respect to the input," in *Proceedings of the 4rd IFAC Workshop on Lagrangian and Hamiltonian Methods for Nonlinear Control*, August 2012, pp. 194–199.

[22] Z. Hao, K. Fujimoto, and Y. Hayakawa, "Optimal trajectory generation for linear systems based on double generating functions," *SICE Journal of Control, Measurement, and System Integration*, 2012, accepted.

[23] H. P. Geering, *Optimal Control with Engineering Applications*, 1st ed. Berlin: Springer, Apr. 2007, pp. 24–35.

[24] Y. Hurmuzlu and T. H. Chang, "Rigid body collisions of a special class of planar kinematic chains," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 5, pp. 964–971, 1992.