Optimal Gait Generation for Legged Robots Based on Variational Symmetry of Hamiltonian Systems

Master Thesis by Satoshi Satoh

Nagoya University
March 2007

Acknowledgement

It would be difficult for me to finish this master thesis without the help and support of many people. I cannot help feeling fortunate in having been given an opportunity to study with them.

First and foremost, I would like to express my deepest respect and gratitude to Associate Professor Kenji Fujimoto for the supervision and his constant support. He has shown great tolerance and patience throughout this work. Valuable discussions with him gave me a lot of things and guided me into several interesting research fields.

I would also like to acknowledge to Professor Yoshikazu Hayakawa for his good advice and comment. They are great help to complete this work. His critical reading of the manuscript provides me with many valuable suggestions.

I would like to thank Dr. Sang-Ho Hyon of ATR/JST. He has collaborated on the researches in this master thesis and gave me a significant advice and encouragement.

I am grateful to Professor Yoji Uno for his technical advice and careful reading of the manuscript.

I would like to thank Research Associate Akira Nakashima for his valuable advice.

Finally, for providing a simultaneous environment for my work, I would like to thank all the staff and students of Intelligent Mechatronics Laboratory of Nagoya University.

Contents

Chapter 1

Introduction

Recently, the walking control has become active research area. As the technology for walking robots evolves, an optimization problem of gaits with respect to the energy consumption becomes important increasingly. In this point of view, *passive dynamic walking* studied by McGeer [10] attracts attention. The gait is stable and periodic on a gentle slope and it is generated with no actuation of any kind, i.e., powered only by gravity. Behavior analysis of passive walkers were studied by several researchers, [11, 12]. In [6, 15, 3], gait generation methods based on passive dynamic walking, i.e., designing appropriate feedback control systems so that the closed loop systems behave like passive walkers are also proposed. Other legged robots which have passive gaits are also considered. For example, a spring-driven one-legged hopping robot called a *passive running robot* can run with zero control input under appropriate initial conditions [16]. This implies that the gait can be regarded as a passive one on a horizontal plane.

In the meanwhile, A novel iterative learning control (ILC) method was proposed in [5]. It is based on a property of Hamiltonian systems called *variational symmetry*. A conventional iterative learning control method studied in [2] can generate a feedforward input to achieve a trajectory tracking control by iteration of experiments without using precise knowledge of the plant model. On the other hand, the novel method has a big advantage that it can achieve not only the trajectory tracking control but also a class of optimal one.

Our objective is to generate optimal walking gait trajectories for legged robots via iterative learning control. Our approach is robust over modeling error since it does not require precise knowledge of the plant model. In this paper, three novel techniques with respect to the iterative learning control are proposed. Firstly, we define a cost function to achieve time symmetric trajectories to generate stable periodic gaits. However, the existing iterative learning control method has a constraint with respect to cost functions. This method can not deal with functionals of velocity of typical mechanical systems. Since velocity has direct effect to the mechanical energy, it is important to check the behavior of it for gait generation problem. Here we propose a technique to relax the limitation with respect to cost functions by employing a pseudo adjoint of the time derivative operator. We consider the passive running robot and succeed in generating *sub-optimal* gait trajectories minimizing the L_2 norm of the control input. But they are not optimal in a sense that the running gaits with zero input are not obtained. This is because the proposed method can not take into account the variation of the initial condition. Generally, passive gaits depend on their initial conditions. So only the input iteration can not generate such gaits. We propose a novel algorithm to generate *optimal* gait trajectories by employing an update law for the initial condition as well as that for the feedforward input. We generate passive running gait trajectory as an optimal solution applying this framework. The foregoing two techniques are in the authors' former result [13]. We also apply this proposed scheme to the *compass gait biped* [7], which is a simple planar biped robot and which has passive walking gaits [14]. Symmetric trajectories are one of the sufficient conditions for running gaits of the passive running robot [9]. However, these are not always sufficient ones for other biped robots including the compass gait biped. In this point of view, we propose a novel framework to generate periodic trajectories which satisfy one of the necessary conditions for periodic gaits. Generally, in walking motion, there is a collision between a leg and the ground. Our novel framework can deal with such discrete state transitions without using the numerical transition model. Transition equations derived by the conservation law of angular momentum are often used in walking analysis. However, this law does not hold exactly with real robots. Applying this method, we generate passive walking gait of the compass gait biped on a slope. Furthermore, an optimal gait trajectories on the level ground can be generated. Numerical simulations demonstrate the effectiveness of the proposed framework and it is expected that our method is valid for general walking robots.

Chapter 2

Iterative learning control based on variational symmetry

This chapter refers to the iterative learning control (ILC) method based on variational symmetry in [5] briefly, which is the important basis of this study. The objective of the method is to solve a class of optimal control problems by iteration of laboratory experiments. Since it is based on optimal control, the adjoint of the target system has to be taken. Generally, the precise model of the target system is needed to construct its adjoint. However, this method does not require it in taking advantage of a property on the self-adjoint related structure of the variationals of Hamiltonian systems called *variational symmetry*.

2.1 Variational symmetry of Hamiltonian systems

Consider a Hamiltonian system with dissipation Σ with a controlled Hamiltonian $H(x, u, t)$ described as $(x^1, y) = \Sigma(x^0, u)$:

$$\begin{cases} \dot{x} = (J - R) \dfrac{\partial H(x,u,t)}{\partial x}^{\mathrm{T}}, \quad x(t^0) = x^0 \\ y = -\dfrac{\partial H(x,u,t)}{\partial u}^{\mathrm{T}} \\ x^1 = x(t^1) \end{cases} . \tag{2.1}$$

Here $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}^r$ describe the state, the input and the output, respectively. The structure matrix $J \in \mathbb{R}^{n \times n}$ and the dissipation matrix $R \in \mathbb{R}^{n \times n}$ are skew-symmetric and symmetric positive semi-definite, respectively. The matrix R represents dissipative elements, for example, frictions and resistances.

Here let us recall Fréchet derivative of nonlinear operators.

Definition 2.1 Consider an operator $f : X \to Y$ with Banach spaces X and Y. f is said to be *Fréchet differentiable* at $x \in X$ if there exists an operator $\mathrm{d}f : X \times X \to Y$ such that $\mathrm{d}f(x)(\xi)$ is linear in ξ and that

$$\lim_{\|\xi\|_X \to 0} \frac{\|f(x+\xi) - f(x) - \mathrm{d}f(x)(\xi)\|_Y}{\|x\|_X} = 0.$$

Under these circumstances, $\mathrm{d}f(x)(\cdot)$ is called the *Fréchet derivative* of f at x.

For the system in (2.1), the following theorem holds. This property is called *variational symmetry* of Hamiltonian control systems.

Theorem 2.1 *[5] Consider the Hamiltonian system in* (2.1)*. Suppose that J and R are constant and that there exists a nonsingular matrix $T \in \mathbb{R}^{n\times n}$ satisfying*

$$\begin{aligned} J &= -TJ\,T^{-1} \\ R &= TR\,T^{-1} \\ \frac{\partial^2 H(x,u,t)}{\partial(x,u)^2} &= \begin{pmatrix} T & 0 \\ 0 & I \end{pmatrix} \frac{\partial^2 H(x,u,t)}{\partial(x,u)^2} \begin{pmatrix} T^{-1} & 0 \\ 0 & I \end{pmatrix}. \end{aligned} \tag{2.2}$$

Then the Fréchet derivative of Σ is described by another linear Hamiltonian system $(x_v^1, y_v) = \mathrm{d}\Sigma((x^0, u), (x_v^0, u_v))$:

$$\begin{cases} \dot{x} = (J-R)\dfrac{\partial H(x,u,t)}{\partial x}^{\mathrm{T}}, & x(t^0) = x^0 \\ \dot{x}_v = (J-R)\dfrac{\partial H_v(x,u,x_v,u_v,t)}{\partial x_v}^{\mathrm{T}}, & x_v(t^0) = x_v^0 \\ y_v = -\dfrac{\partial H_v(x,u,x_v,u_v,t)}{\partial u_v}^{\mathrm{T}} \\ x_v^1 = x_v(t^1) \end{cases} \tag{2.3}$$

with a controlled Hamiltonian $H_v(x,u,x_v,u_v,t)$

$$H_v(x,u,x_v,u_v,t) = \frac{1}{2}\begin{pmatrix} x_v \\ u_v \end{pmatrix}^{\mathrm{T}} \frac{\partial^2 H(x,u,t)}{\partial(x,u)^2} \begin{pmatrix} x_v \\ u_v \end{pmatrix}.$$

Furthermore, the adjoint of the variational system with zero initial state $u_a \mapsto y_a = (\mathrm{d}\Sigma^{x^0}(u))^*(u_a)$ is given by

$$\begin{cases} \dot{x} = (J-R)\dfrac{\partial H(x,u,t)}{\partial x}^{\mathrm{T}} \\ \dot{\bar{x}}_v = -(J-R)\dfrac{\partial H_v(x,u,\bar{x}_v,u_a,t)}{\partial \bar{x}_v}^{\mathrm{T}} \\ y_a = -\dfrac{\partial H_v(x,u,\bar{x}_v,u_a,t)}{\partial u_a}^{\mathrm{T}} \end{cases} \tag{2.4}$$

with the initial state $x(t^0) = x^0$ and the terminal state $\bar{x}_v(t^1) = 0$. Suppose moreover that $J-R$ is nonsingular. Then the adjoint $(x_a^1, u_a) \mapsto (x_a^0, y_a) = (\mathrm{d}\Sigma(x^0, u))^*(x_a^1, u_a)$ is given by the same state-space realization (2.4) *with the initial state $x(t^0) = x^0$ and the terminal state $\bar{x}_v(t^1) = -(J-R)T\,x_a^1$ and $x_a^0 = -T^{-1}(J-R)^{-1}\bar{x}_v(t^0)$.*

Remark 2.2 This theorem reveals that the variational system and its adjoint of a Hamiltonian system in the form (2.1) have almost the same state-space realizations. This means that the input-output mapping of the adjoint can be calculated by the input-output data of the original system as

$$\begin{aligned}\pi_U \circ (\mathrm{d}\Sigma(x^0, u))^*(\xi, v) &= \mathcal{R} \circ \pi_Y \circ (\mathrm{d}\Sigma(x^0, u))\big(-(J-R)T\xi, \mathcal{R}(v)\big) \\ &\approx \mathcal{R} \circ \pi_Y \circ \Big(\Sigma\big(x^0 - (J-R)T\xi, u + \mathcal{R}(v)\big) - \Sigma(x^0, u)\Big)\end{aligned} \tag{2.5}$$

provided appropriate boundary conditions are selected, where $\pi_{(\cdot)}$ denotes the projection operator onto $(\cdot)$ and $\mathcal{R}$ is the time reversal operator defined by

$$\mathcal{R}(u)(t - t^0) = u(t^1 - t), \quad \forall t \in [t^0, t^1]. \tag{2.6}$$

Let us take an example for the boundary conditions. the input u is given such that the time history of the Hessian of the Hamiltonian with respect to (x, u) is symmetrical with respect to the middle of the time interval $(t^0 + t^1)/2$, i.e.,

$$\frac{\partial^2 H(x,u,t)}{\partial (x,u)^2}(t - t^0) = \frac{\partial^2 H(x,u,t)}{\partial (x,u)^2}(t^1 - t), \quad \forall t \in [t^0, t^1]. \tag{2.7}$$

Then $\mathrm{d}\Sigma$ has a self-adjoint sate-space realization. A useful alternative condition for mechanical systems is discussed in [5]. A procedure in case where the condition (2.7) does not hold is considered in [4].

This property is utilized for solving the optimal control problems in which the adjoint operator plays an important role.

2.2 Optimal control via iterative learning

This section explains how to apply the property of the variational systems of Hamiltonian systems referred in Section 2.1 to iterative learning control.

Let us consider the system $\Sigma : X \times U \to X \times Y$ in (2.1) and a cost function $\Gamma : X^2 \times U \times Y \to \mathbb{R}$ with Hilbert spaces X, U and Y. Typically, $X = \mathbb{R}^n$, $U = L_2^m[t^0, t^1]$ and $Y = L_2^r[t^0, t^1]$. The objective is to find the optimal input minimizing the cost function $\Gamma(x^0, u, x^1, y)$. Note that the Fréchet derivative of $\Gamma(x^0, u, x^1, y)$ is $\mathrm{d}\Gamma(x^0, u, x^1, y)$ and the cost function Γ is described by $\Gamma(x^0, u, x^1, y) = \Gamma((x^0, u), \Sigma(x^0, u))$ by $(x^1, y) = \Sigma(x^0, u)$. Here we can calculate

$$\begin{aligned}&\mathrm{d}\left(\Gamma(x^0, u, x^1, y)\right)(\mathrm{d}x^0, \mathrm{d}u, \mathrm{d}x^1, \mathrm{d}y) \\ &= \mathrm{d}\Gamma((x^0, u), \Sigma(x^0, u))\left((\mathrm{d}x^0, \mathrm{d}u), \mathrm{d}\Sigma(x^0, u)(\mathrm{d}x^0, \mathrm{d}u)\right) \\ &= \langle \Gamma'((x^0, u), \Sigma(x^0, u)), \begin{pmatrix} \mathrm{id}_{X\times U} \\ \mathrm{d}\Sigma(x^0, u)\end{pmatrix}(\mathrm{d}x^0, \mathrm{d}u)\rangle_{X^2\times U\times Y} \\ &= \langle \left(\mathrm{id}_{X\times U}, (\mathrm{d}\Sigma(x^0, u))^*\right) \Gamma'(x^0, u, x^1, y), (\mathrm{d}x^0, \mathrm{d}u)\rangle_{X\times U},\end{aligned} \tag{2.8}$$

where id represents the identity mapping. Well-known Riesz's representation theorem and the linearity of Fréchet derivative guarantee that there exists an operator $\Gamma'(x^0, u, x^1, y)$ as above. Therefore, if the adjoint $(\mathrm{d}\Sigma(x^0, u))^*$ is available, we can reduce the cost function Γ down at least to a local minimum by an iteration law with a $K_{(i)} > 0$.

$$u_{(i+1)} = u_{(i)} - K_{(i)}\,(0_{UX}, \mathrm{id}_U)\left(\mathrm{id}_{X\times U},\ (\mathrm{d}\Sigma(x^0_{(i)}, u_{(i)}))^*\right)\Gamma'(x^0_{(i)}, u_{(i)}, x^1_{(i)}, y_{(i)}) \tag{2.9}$$

Here i denotes the i-th iteration in laboratory experiment.

The results in the previous section enable one to execute this procedure without using the parameters of the original operator Σ by the relation (2.5), provided Σ is a Hamiltonian system and the boundary conditions are selected appropriately.

Chapter 3

Extension of ILC for time derivatives

Let us recall that there is a constraint with respect to cost functions in the iterative learning control method in [5]. For the system Σ in (2.1), the output y is uniquely defined by the definition of the input u. The possible choice of the optimal control type cost function used in the iterative learning control is a functional of u and y, and it is not possible to choose a functional of $\dot{y}$ the time derivative of the output. However, the signal $\dot{y}$ often plays an important role in control systems and, particularly, it is important to check the behavior of $\dot{y}$ for the gait trajectory generation problem. In this chapter, we extend the iterative learning control method referred in the previous chapter to take the time derivative $\dot{y}$ into account.

3.1 Pseudo adjoint of the time derivative operator

Here we investigate a pseudo adjoint of the time derivative operator in [13] to take account of the time derivative of the output signal $\dot{y}$ in the iterative learning control procedure. Consider a differentiable signal $\xi \in L_2[t^0, t^1]$ and an operator $D(\cdot)$ which maps the signal $\xi(t)$ into its time derivative is defined as the time derivative operator.

$$D(\xi)(t) := \frac{\mathrm{d}\xi(t)}{\mathrm{d}t} \tag{3.1}$$

Let us provide the following lemma to define the pseudo adjoint of the time derivative operator.

Lemma 3.1 *[13] Consider the signal $\xi(t)$ defined above and another differentiable signal $\eta \in L_2[t^0, t^1]$. Suppose that the signal $\xi(t)$ satisfies the following condition*

$$\xi(t^0) = \xi(t^1) = 0. \tag{3.2}$$

Then the following equation holds.

$$\langle \eta, D(\xi) \rangle_{L_2} = \langle -D(\eta), \xi \rangle_{L_2} \tag{3.3}$$

Proof. Consider the inner product of η and $D(\xi)$. Let us calculate that

$$\begin{aligned}\langle \eta, D(\xi)\rangle_{L_2} &= \int_{t^0}^{t^1} \eta(t)^{\mathrm{T}} \frac{\mathrm{d}\xi(t)}{\mathrm{d}t}\mathrm{d}t \\ &= \Big[\eta(t)^{\mathrm{T}}\xi(t)\Big]_{t^0}^{t^1} - \int_{t^0}^{t^1} \frac{\mathrm{d}\eta(t)}{\mathrm{d}t}^{\mathrm{T}} \xi(t)\mathrm{d}t.\end{aligned}$$

Here $\xi(t)$ satisfies the condition (3.2), therefore $\Big[\eta(t)^{\mathrm{T}}\xi(t)\Big]_{t^0}^{t^1} = 0$ holds and down to

$$\begin{aligned}\langle \eta, D(\xi)\rangle_{L_2} &= -\int_{t^0}^{t^1} \frac{\mathrm{d}\eta(t)}{\mathrm{d}t}^{\mathrm{T}} \xi(t)\mathrm{d}t \\ &= \langle -D(\eta), \xi\rangle_{L_2}.\end{aligned} \tag{3.4}$$

Then (3.4) implies (3.3). □

This lemma implies

$$D^* = -D$$

for a certain class of input signals.

3.2 Application to the iterative learning control

Here we take the following cost function $\Gamma(\dot{y})$ to illustrate the proposed method

$$\Gamma(\dot{y}) = \frac{1}{2}\int_{t^0}^{t^1} \Big((\dot{y}(t) - \dot{y}^d(t))^{\mathrm{T}}\Lambda_{\dot{y}}(\dot{y}(t) - \dot{y}^d(t))\Big)\mathrm{d}t. \tag{3.5}$$

Here $\dot{y}^d$ is a differentiable signal as a desired velocity satisfying $\dot{y}^d \in L_2^r[t^0, t^1]$ and $\Lambda_{\dot{y}} \in \mathbb{R}^{r\times r}$ is a positive definite matrix. Let us consider the Hamiltonian system in (2.1) and suppose that the following assumption holds.

Assumption 3.1 Following conditions always hold
$\mathrm{d}y(t^0) = 0$ and $\mathrm{d}y(t^1) = 0$.

Under this assumption, we will derive an iterative learning control method which can handle the time derivative of the output function $\dot{y}$ In the iterative learning control, it is assumed that all the initial conditions are same in each laboratory experiment in general. Therefore the condition $\mathrm{d}y(t^0) = 0$ always holds. But the other one $\mathrm{d}y(t^1) = 0$ does not always hold. In order to let the latter condition $\mathrm{d}y(t^1) = 0$ hold approximately, we can employ an optimal control type cost function such as $\int_{t^1-\epsilon}^{t^1} \|y(t) - y^d(t)\|^2\mathrm{d}t$ with a small constant $\epsilon > 0$ as in [4].

Suppose that the output y fulfills Assumption 3.1. Then we have

$$\mathrm{d}(\Gamma(\dot{y}))(\mathrm{d}\dot{y}) = \langle \Lambda_{\dot{y}}(\dot{y} - \dot{y}^d), \mathrm{d}\dot{y}\rangle_{L_2}$$

The existing result [5] can not directly apply to this cost function (3.5) because it contains $\dot{y}$. Here let us rewrite $\dot{y}$ as $\dot{y} = D(y)$ with the time derivative operator $D(\cdot)$ defined by Equation (3.1). Then we have

$$\mathrm{d}\dot{y} = \mathrm{d}D(y)(\mathrm{d}y). \tag{3.6}$$

Note that the time derivative operator is linear, we obtain $\mathrm{d}\dot{y} = D(\mathrm{d}y)$. Assumption 3.1 and (3.3) imply

$$\begin{aligned} \mathrm{d}(\Gamma(\dot{y}))(\mathrm{d}\dot{y}) &= \langle \Lambda_{\dot{y}}(\dot{y} - \dot{y}^d), D(\mathrm{d}y)\rangle_{L_2} \\ &= \langle -D\big(\Lambda_{\dot{y}}(\dot{y} - \dot{y}^d)\big), \mathrm{d}y\rangle_{L_2} \\ &= \langle (\mathrm{d}\Sigma(u))^* \big(-D(\Lambda_{\dot{y}}(\dot{y} - \dot{y}^d))\big), \mathrm{d}u\rangle_{L_2} \end{aligned} \tag{3.7}$$

As we mentioned above, this method allows one to obtain an iterative learning control method in which a cost function consisting of $\dot{y}$.

Chapter 4

Optimal gait generation for a one-legged running robot

In this chapter, we propose a novel algorithm to generate optimal gait trajectories based on the iterative learning control and the proposed scheme is applied to a one-legged hopping robot called a *passive running robot* [16]. Let us note that this robot has passive running gaits with zero control input under appropriate initial conditions.

Firstly, a cost function to generate a *symmetric gait* is proposed for guaranteeing periodic running of the running robot without a fall. In fact, [9] implies that symmetric of the trajectory is a sufficient condition for running gaits of this robot. At this stage of our proposed method, *sub-optimal* gait trajectories minimizing the L_2 norm of the control input under a given initial condition can be generated.

Secondly, we derive an update law for the initial conditions and combine this with the learning procedure. This algorithm allows one to obtain an *optimal* gait in a sense that corresponding input coincides with zero, i.e., passive running gait.

4.1 Description of the plant

Let us consider the passive running robot depicted in Figure 4.1. Here the body and the leg have mass m_b and m_l and moment of inertia J_b and equivalent leg inertia J_l, respectively. Let us also define the control force of the leg ρ and the control torque of the hip joint τ. Table 5.1 shows the physical parameters.

Here the stance time represents the time interval during the stance phase and the flight time is defined in a similar way. One locomotion cycle is illustrated in Figure 4.2. It consists of the *stance phase*, where the leg touches the ground and the leg spring is compressed, and the *flight phase*, where the leg is above the ground and the robot traverses a ballistic trajectory. Furthermore, some assumptions are assumed on this robot. An important one is as follows.

Assumption 4.1 The foot does not bounce back nor slip on the ground (inelastic impulsive impact).

See [1, 9] for the rest of them.

In the stance phase, let us define the generalized coordinate $q := (r, \theta, \phi)^{\mathrm{T}} \in \mathbb{R} \times \mathbb{S} \times \mathbb{S}$, the generalized momentum $p := (p_r, p_\theta, p_\phi)^{\mathrm{T}} \in \mathbb{R}^3$, input $u := (\rho, \tau)^{\mathrm{T}} \in \mathbb{R}^2$ and the inertia matrix

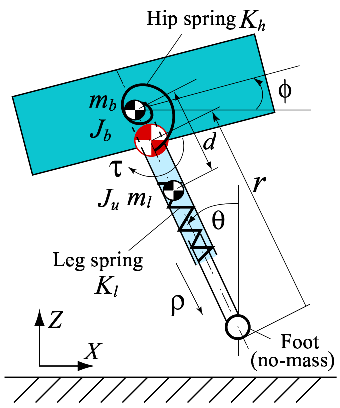

Figure 4.1: Model of the passive running robot

Table 4.1: Parameters

Notation	Meaning	Unit
r_0	natural leg length	m
m	total mass	kg
g	gravity acceleration	$\mathrm{m/s^2}$
K_l	leg spring stiffness	N/m
K_h	hip spring stiffness	Nm/rad
T_s	stance time	s
T_f	flight time	s

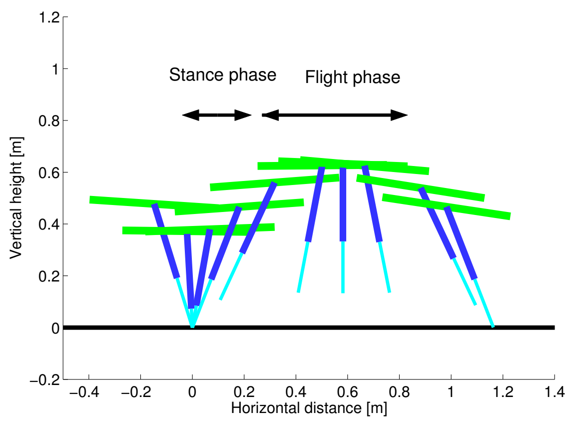

Figure 4.2: Locomotion phases during one cycle

$M(q) \in \mathbb{R}^{3\times 3}$. Then, the dynamics of this robot is described by a Hamiltonian system in (2.1) with the Hamiltonian

$$H(q,p,u) = \frac{1}{2}p^{\mathrm{T}}M(q)^{-1}p - mgr(1-\cos\theta) + \frac{1}{2}K_l(r-r_0)^2 + \frac{1}{2}K_h(\theta-\phi)^2 - u^{\mathrm{T}}\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & -1\end{pmatrix}q.$$

Let us consider the dynamics in the flight phase as below

$$\begin{cases} \ddot{x} = 0 \\ \ddot{z} = -g \\ J_l\ddot{\theta} + J_b\ddot{\phi} = 0 \\ J_b\ddot{\phi} = K_h(\theta-\phi) - \tau_f \end{cases} . \tag{4.1}$$

Here the variables x and z represent the horizontal and vertical positions of the center of mass and τ_f is the control torque.

4.2 Generation of symmetric gaits via iterative learning control

This section sets the control problem to get the periodic gait based on [8].

The leg angle θ is the most important state variable in controlling running gait, because it has a direct effect to avoid falling. However, it is difficult to control the variable θ in the stance phase, since this robot has no foot and ankle torque is not available. Therefore as in [8], we apply zero input in the stance phase, i.e., $\rho \equiv \tau \equiv 0$ and try to control the variable θ in the flight phase to get the periodic gait.

We let $t^0 = 0$ for simplicity in what follows. Let us define the desired values of θ and $\dot{\theta}$ as

$$\theta^d := \theta\big|_{t=T_s+T_f} = -\theta\big|_{t=T_s} \tag{4.2}$$

$$\dot{\theta}^d := \dot{\theta}\big|_{t=T_s+T_f} = \dot{\theta}\big|_{t=T_s}. \tag{4.3}$$

As for the model mentioned in Section 4.1, energy dissipation occurs at the touchdown. Let E_- and E_+ represent the energies just before the touchdown and just after the touchdown. Then the variation of the energy between them ΔE can be calculated as below from [9]

$$\Delta E := E_- - E_+ = \frac{mJ_l}{2(J_l + mr_0^2)}\mu_-^2, \tag{4.4}$$

where μ_- is defined as follows and is called the energy dissipation coefficient

$$\mu_- := v_{x\_}cos\theta_- + v_{z\_}sin\theta_- + \frac{r_0}{J_l + mr_0^2}p_{\theta\_}. \tag{4.5}$$

Here $v_{x\_}$ and $v_{z\_}$ represent the velocity of the center of mass. Suppose that the condition (4.6) holds at the touchdown

$$\Delta E = 0. \tag{4.6}$$

This implies that there is no energy transfer except for the control input. If the total mechanical energy is completely preserved, it is expected that a periodic gait trajectories are autonomously generated. In fact, [9] implies that the condition (4.6) is satisfied if the control objects (4.2) and (4.3) are achieved. The initial condition is appropriately chosen according to [9].

In [9], dead-beat control is applied to this problem. It works well, but it requires precise knowledge of the plant system. Here we try to use iterative learning control based on variational symmetry with a special cost function given as follows. It will generate an optimal flow in the flight phase without using precise knowledge of the system.

Now we propose a cost function as

$$\Gamma(\theta, \dot{\theta}, u) := \frac{K_\theta}{2}\|\theta - (-\mathcal{R}(\theta))\|_{L_2}^2 + \frac{K_{\dot{\theta}}}{2}\|\dot{\theta} - \mathcal{R}(\dot{\theta})\|_{L_2}^2 + \frac{K_u}{2}\|u\|_{L_2}^2 \tag{4.7}$$

where K_θ, $K_{\dot{\theta}}$ and K_u represent appropriate positive constants. $\mathcal{R}$ is the time-reversal operator as defined in (2.6). The first term in the right hand side of (4.7) makes Assumption 3.1 approximately hold. It is expected that we can generate an optimal trajectory such that it satisfies (4.2) and (4.3) while minimizing the L_2 norm of the control input. Furthermore, there is no energy transfer except for the control input.

Let us recall the fact that gait trajectories are essentially periodic. However, ILC can not generate a periodic trajectory. We connect the stance flow generated from (4.7). Take this connected trajectory as an one period of a periodic gait trajectory. Therefore if (4.6) holds, we can generate an optimal periodic trajectory.

We define the input $u = \tau_f$ and rewrite the dynamics of θ in the flight phase. Under the initial condition selected as in [9], we can remove the dynamics of θ from the dynamics in (4.1) as below

$$\begin{pmatrix} \dot{q_\theta} \\ \dot{p_\theta} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} \dfrac{\partial H(q_\theta, p_\theta, u)}{\partial q_\theta}^{\mathrm{T}} \\ \dfrac{\partial H(q_\theta, p_\theta, u)}{\partial p_\theta}^{\mathrm{T}} \end{pmatrix}$$

$$y = -\frac{\partial H(q_\theta, p_\theta, u)}{\partial u}^{\mathrm{T}} = q_\theta = \theta, \tag{4.8}$$

where $q_\theta := \theta$, $p_\theta := J_l\dot{\theta}$ and a controlled Hamiltonian

$$H(q_\theta, p_\theta, u) := \frac{1}{2J_l}p_\theta^2 + \frac{1}{2}\frac{K_h(J_b + J_l)}{J_b}q_\theta^2 - q_\theta u.$$

Now let us calculate the iteration law with respect to the cost function (4.7). The following lemma is provided to equip the adjoint of the time-reversal operator.

Lemma 4.1 *Consider the time-reversal operator $\mathcal{R}(\cdot)$ defined in* (2.6)*, then the following equation holds*

$$(\mathcal{R}(\cdot))^* = \mathcal{R}(\cdot) \tag{4.9}$$

Proof. Consider signals ξ and $\eta \in L_2[t^0, t^1]$. Let us calculate the inner product of ξ and $\mathcal{R}(\eta)$ that

$$\begin{aligned}\langle \xi, \mathcal{R}(\eta)\rangle_{L_2} &= \int_{t^0}^{t^1} \xi(t)^{\mathrm{T}}\eta(t^0+t^1-t)\mathrm{d}t \\ &= \int_{t^0}^{t^1} \xi(t^0+t^1-s)^{\mathrm{T}}\eta(s)\mathrm{d}s \\ &= \langle \mathcal{R}(\xi), \eta\rangle_{L_2}, \end{aligned} \tag{4.10}$$

where $s = t^0 + t^1 - t$. This implies (4.9). □

Since the time-reversal operator is linear, we obtain $\mathrm{d}\mathcal{R}(\cdot) = \mathcal{R}(\mathrm{d}(\cdot))$. By this equation, Equations (3.3) and (4.9), we can calculate Fréchet derivative of (4.7) that

$$\begin{aligned}\mathrm{d}(\Gamma(\theta,\dot\theta,u)) &= \langle K_\theta(\theta+\mathcal{R}(\theta)), \mathrm{d}\theta+\mathcal{R}(\mathrm{d}\theta)\rangle_{L_2} + \langle K_{\dot\theta}(\dot\theta-\mathcal{R}(\dot\theta)), \mathrm{d}\dot\theta-\mathcal{R}(\mathrm{d}\dot\theta)\rangle_{L_2} \\ &\quad + \langle K_u u, \mathrm{d}u\rangle_{L_2} \\ &= \langle K_\theta(\mathrm{id}+\mathcal{R})(\theta+\mathcal{R}(\theta)), \mathrm{d}\theta\rangle_{L_2} + \langle K_{\dot\theta}(\mathrm{id}-\mathcal{R})(\dot\theta-\mathcal{R}(\dot\theta)), \mathrm{d}\dot\theta\rangle_{L_2} \\ &\quad + \langle K_u u, \mathrm{d}u\rangle_{L_2} \\ &= \langle 2K_\theta(\mathrm{id}+\mathcal{R})(\theta) - 2K_{\dot\theta}(\mathrm{id}-\mathcal{R})(\ddot\theta), \mathrm{d}\theta\rangle_{L_2} + \langle K_u u, \mathrm{d}u\rangle_{L_2} \\ &= \langle K_u u + (\mathrm{d}\Sigma(u))^*\big(2K_\theta(\mathrm{id}+\mathcal{R})(\theta) - 2K_{\dot\theta}(\mathrm{id}-\mathcal{R})(\ddot\theta)\big), \mathrm{d}u\rangle_{L_2}\end{aligned}$$

Therefore, the steepest descent method implies that we should change the input u such that

$$\mathrm{d}u = -K\Big(K_u u + (\mathrm{d}\Sigma(u))^*\big(2K_\theta(\mathrm{id}+\mathcal{R})(\theta) - 2K_{\dot\theta}(\mathrm{id}-\mathcal{R})(\ddot\theta)\big)\Big), \tag{4.11}$$

where K is an appropriate positive gain. Here let us confirm whether Theorem 2.1 holds.

Remark 4.2 Consider the dynamics of θ in the flight phase in (4.8). By Equations (2.1) and (4.8), we have

$$J = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \quad R = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

Here we choose a nonsingular matrix T in Theorem 2.1 as

$$T = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{4.12}$$

Then, matrices J, R and T satisfy Theorem 2.1. The Hessian of the Hamiltonian in (4.8) is

$$\frac{\partial^2 H(q_\theta, p_\theta, u)}{\partial(q_\theta, p_\theta, u)^2} = \begin{pmatrix} \dfrac{K_h(J_b+J_l)}{J_b} & 0 & -1 \\ 0 & \dfrac{1}{J_l} & 0 \\ -1 & 0 & 0 \end{pmatrix}. \tag{4.13}$$

Using Equations (4.12) and (4.13), we can easily prove that Theorem 2.1 holds.

Now applying Equation (2.5) to Equation (4.11), we can derive the iteration law provided that the initial input $u_{(0)}$ is equivalent to zero as below

$$u_{(2i+1)} = u_{(2i)} + \epsilon_{(i)} \mathcal{R}\Big(2K_\theta(\mathrm{id} + \mathcal{R})(\theta_{(2i)}) - 2K_{\dot\theta}(\mathrm{id} - \mathcal{R})(\ddot\theta_{(2i)})\Big) \quad (4.14)$$

$$u_{(2i+2)} = u_{(2i)} - K_{(2i)}\Big(K_u u_{(2i)} + \frac{1}{\epsilon_{(i)}}\mathcal{R}(\theta_{(2i+1)} - \theta_{(2i)})\Big). \quad (4.15)$$

Here $\epsilon_{(\cdot)}$ represents a sufficiently small constant. The pair of iteration laws (4.14) and (4.15) implies that this learning procedure needs two steps laboratory experiments, that is, it requires two experiments to execute a single update step in the steepest decent method. In the (2i+1)-th iteration, we can get the output signal of $\Sigma\big(x^0 - (J-R)T\xi, \bar u + \mathcal{R}(v)\big)$ in (2.5) and then we can calculate the input and output signals of $(\mathrm{d}\Sigma)^*$ using Equation (2.5). The input for the (2i+2)-th iteration is generated by (2.9) with these signals.

4.3 Generation of passive running gait

The iteration procedure mentioned in the previous section can generate a *sub-optimal* symmetric gait minimizing the L_2 norm of the control input. But this method can not generate an *optimal* one in a sense that the corresponding input does not coincide with zero, though the passive running robot depicted in Figure 4.1 can run with zero input under appropriate initial conditions. In the existing results on iterative learning control, it is assumed that all the initial conditions are the same. However, passive gaits depend on their initial conditions. It means that only the input iteration can not generate such gaits. In this section, we derive a novel update law for the initial condition to generate an *optimal* passive gait.

Since the state space is 6 dimensional, the initial condition to be determined is $(\dot x^0, \dot z^0, \theta^0, \dot\theta^0, \phi^0, \dot\phi^0)$. Here we select $\dot x^0$ and θ^0 as free parameters and let the rest $(\dot z^0, \dot\theta^0, \phi^0, \dot\phi^0)$ be calculated according to $\dot x^0$ and θ^0 as in [9]. In this section, let us update one of the free parameters θ^0 as well as the control input u to achieve the optimal initial conditions under which passive running gait is generated. The other free parameter $\dot x^0$ is not determined, so we can select the horizontal velocity of passive running gait by choosing it appropriately. A more detailed procedure is explained below. Here let ${X^0}_{(i)} \in \mathbb{R}^6$ denote the initial condition in (2i+1)-th and (2i+2)-th Steps.

Step 0 **:** Set the constant positive parameter δ, where δ denotes the desired value of the L_2 norm of the control input. We let δ be sufficiently small. Set the initial free parameters $\dot x^0$ and ${\theta^0}_{(1)}$ and calculate the other initial conditions according to [9]. Then we get the first initial condition ${X^0}_{(1)}$.

Step $2i+1$ **:** With ${X^0}_{(i)}$, executes the (2i+1)-th laboratory experiment via the iteration law of (4.14), then goto Step 2i+2.

Step $2i+2$ **:** With ${X^0}_{(i)}$, executes the (2i+2)-th laboratory experiment via the iteration law of (4.15). If $\|u_{(2i+2)}\|_{L_2} \le \delta$, the procedure terminates. Here $\delta > 0$ is a prescribed sufficiently small constant.

Otherwise, update ${\theta^0}_{(i)}$ by the update law

$$\theta^0_{(i+1)} = \theta^0_{(i)} - K_{\theta^0(i)}\, p_{\theta_v(i)}\Big|_{t=t^1}, \tag{4.16}$$

where K_{θ^0} is an appropriate positive gain and p_{θ_v} is the variation of the p_θ in (4.8). We get the (i+1)-th initial condition ${X^0}_{(i+1)}$ and goto Step (2i+3).

Let us derive the update law (4.16). In Equation (2.8), we write

$$\Gamma'(x^0, u, x^1, y) =: (\Gamma'_{x^0}, \Gamma'_u, \Gamma'_{x^1}, \Gamma'_y)^{\mathrm{T}}.$$

Here we can calculate

$$\begin{aligned}\mathrm{d}\left(\Gamma((x^0, u), \Sigma(x^0, u))\right)(\mathrm{d}x^0, \mathrm{d}u) = {} & \langle \Gamma'_{x^0} + \pi_{\mathbb{R}^n} \circ (\mathrm{d}\Sigma(x^0, u))^*(\Gamma'_{x^1}, \Gamma'_y), \mathrm{d}x^0\rangle \\ & + \langle \Gamma'_u + \pi_U \circ (\mathrm{d}\Sigma(x^0, u))^*(\Gamma'_{x^1}, \Gamma'_y), \mathrm{d}u\rangle. \end{aligned} \tag{4.17}$$

The steepest decent method implies that we should update the initial condition such that

$$\mathrm{d}x^0 = -K_{x^0}\Big(\Gamma'_{x^0} + \pi_{\mathbb{R}^n} \circ (\mathrm{d}\Sigma(x^0, u))^*(\Gamma'_{x^1}, \Gamma'_y)\Big) \tag{4.18}$$

where K_{x^0} is an appropriate positive gain.

Suppose that we apply this method to our novel cost function (4.7), we obtain $\Gamma'_{x^0} \equiv 0$ and $\Gamma'_{x^1} \equiv 0$. Here the state at $t = t^0$ is said to be an *initial state* and the one at $t = t^1$ is said to be a *terminal state*. Then let us calculate the initial state of $(\mathrm{d}\Sigma(x^0, u))^*$ with respect to the input $(0, \Gamma'_y)$, that is, $x^0_a := \pi_{\mathbb{R}^n} \circ (\mathrm{d}\Sigma(x^0, u))^*(0, \Gamma'_y)$. As mentioned in Chapter 2, the initial state is calculated as $x^0_a = -T^{-1}(J-R)^{-1}\bar{x}_v(t^0)$, where $\bar{x}_v(t^0)$ is the initial state of the adjoint of the variational system in (2.4). The state of the dynamics in (2.4) is identified with the time-reversal version of that in (2.3), so we can write the initial state of the adjoint of the variational system as $x^0_a = -T^{-1}(J-R)^{-1}x_v(t^1)$, where $x_v(t^1)$ is the terminal state of the variational system in (2.3) under certain circumstances as explained in Remark 2.2. For the dynamics of θ in the flight phase (4.8), $x^0_a = (q_{\theta}{}^0_a, p_{\theta}{}^0_a)^{\mathrm{T}}$ is calculated using Equation (4.12) as follows

$$\begin{aligned}\begin{pmatrix} q_{\theta}{}^0_a \\ p_{\theta}{}^0_a \end{pmatrix} &= -T^{-1}(J-R)^{-1}\begin{pmatrix} q_{\theta_v}(t^1) \\ q_{\theta_v}(t^1) \end{pmatrix} \\ &= -\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}^{-1}\left(\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}\right)^{-1}\begin{pmatrix} q_{\theta_v}(t^1) \\ q_{\theta_v}(t^1) \end{pmatrix} \\ &= \begin{pmatrix} p_{\theta_v}(t^1) \\ q_{\theta_v}(t^1) \end{pmatrix}, \end{aligned} \tag{4.19}$$

where $(q_{\theta_v}(t^1), p_{\theta_v}(t^1))^{\mathrm{T}} =: x_v(t^1)$.

For Equations (4.18) and (4.19), we should update the initial condition θ^0 as

$$\mathrm{d}\theta^0 = -K_{\theta^0}\, p_{\theta_v}(t^1) \tag{4.20}$$

Hence the update law (4.16) is derived immediately.

In this section, we derive a novel update law for initial conditions (4.16) and combining with the learning procedure mentioned in the previous section. The proposed algorithm generates *optimal* passive gait trajectories.

4.4 Numerical examples

We apply the proposed algorithm to the passive running robot considered in Section 4.1. First, we generate a sub-optimal symmetric gait minimizing the L_2 norm of the control input under given initial condition. Then, we generate an optimal passive running gait updating its initial conditions as well as the control inputs.

The concrete values of parameters of the robot are $m_b = 10$ [kg], $m_l = 2$ [kg], $J_b = 0.50$ [kgm^2], $J_l = 0.11$ [kgm^2], $K_h = 10$ [Nm/rad], $K_l = 3000$ [N/m] and $r^0 = 0.5$ [m].

4.4.1 Sub-optimal symmetric gait

We proceed 50 steps of the learning algorithm which means we execute 100 simulations with the initial state $(\dot{x}^0, \dot{z}^0, \theta^0, \dot{\theta}^0, \phi^0, \dot{\phi}^0) = (2, -1.9516, 0.30, -2.6679, -0.0660, 0.5872)$, the gain parameters $K_\theta = 20$, $K_{\dot{\theta}} = 1 \times 10^{-4}$, $K_u = 2 \times 10^{-3}$ and $K_{(\cdot)} = 2$ and $\epsilon_{(\cdot)} = 0.16$.

Figure 4.3 shows that the cost function (4.7) almost decreases at each experiment. This implies that the output trajectory converges to the optimal one smoothly. Since we choose the initial input $u_{(0)}(t) \equiv 0$, Assumption 3.1 does not hold in the beginning. This is probably the reason why the cost function increases in the beginning of the learning procedure. In Figures 4.4 and 4.5, the solid lines show the responses of θ and $\dot{\theta}$ at the last step and the dotted lines depict the initial trajectories corresponding to 0 input. These figures show that both θ and $\dot{\theta}$ converge to the trajectories satisfying (4.2) and (4.3). Figure 4.6 shows that the control input in the last step. These three figures implies that the generated trajectory is the symmetric gait but not passive one. Furthermore, Figure 4.7 exhibits that the variation of the energy at the touchdown ΔE in (4.4) converges to zero automatically.

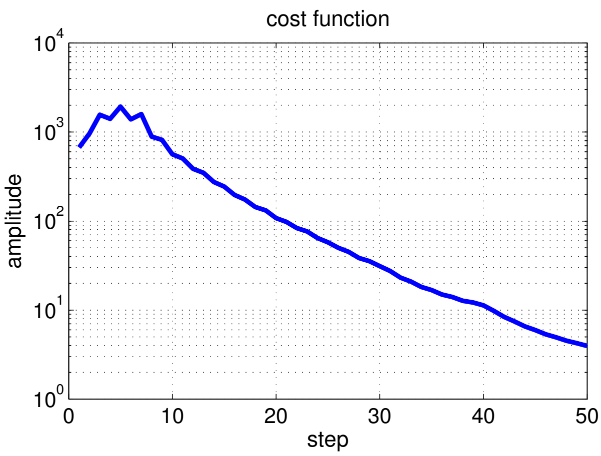

Figure 4.3: Cost function

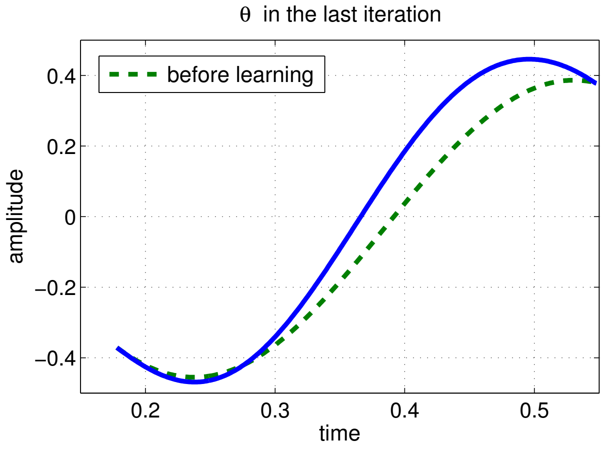

Figure 4.4: Response of θ

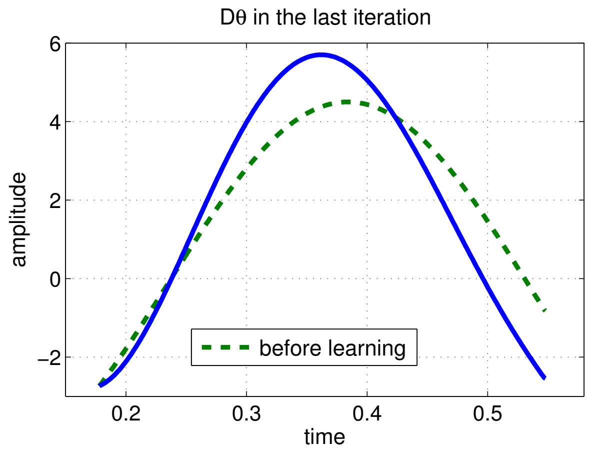

Figure 4.5: Response of $\dot{\theta}$

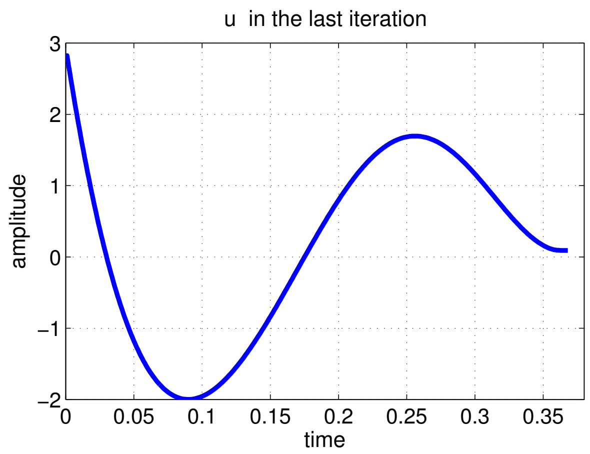

Figure 4.6: Control input in the last iteration

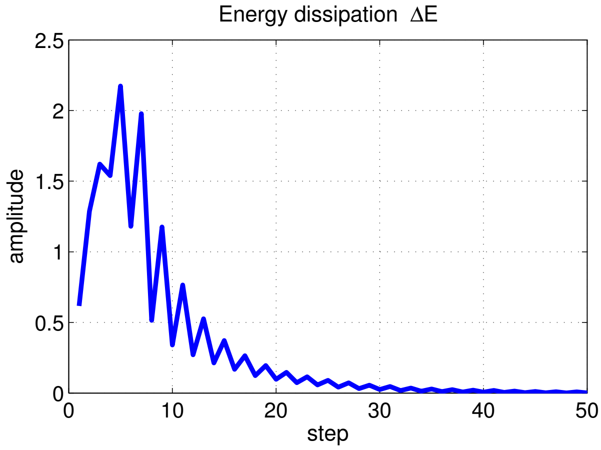

Figure 4.7: Variation of the energy at the touch down ΔE

4.4.2 Passive running gait

We set the parameters mentioned in Section 4.3 as $\delta = 0.025$ and choose the initial condition as the same as that in the previous example, i.e.,

$$X^0_{(1)} = (\dot{x}^0, \dot{z}^0_{(1)}, \theta^0_{(1)}, \dot{\theta}^0_{(1)}, \phi^0_{(1)}, \dot{\phi}_{(1)}) = (2, -1.9516, 0.30, -2.6679, -0.0660, 0.5872).$$

The gain parameters are $K_\theta = 1$, $K_{\dot{\theta}} = 1 \times 10^{-5}$, $K_u = 0.5$ and $K_{(\cdot)} = 2$, $K_{\theta^0(\cdot)} = 0.05$ and $\epsilon_{(\cdot)} = 0.2$. Figure 4.11 shows the procedure terminates at the 34th Step which means we execute 68 simulations.

Figure 4.8 shows the history of the cost function (4.7) along the iteration. It monotonically decreases, which implies that the output trajectory converges to the *optimal* one consequently at each learning experiment. Figures 4.9 and 4.10 show that responses of θ and $\dot{\theta}$ at the last step in the proposed method in solid lines and those of the initial trajectory in dotted lines. These figures imply that both θ and $\dot{\theta}$ converge to the trajectories satisfying the symmetric constraint (4.2) and (4.3). Furthermore Figures 4.11 and 4.12 show that the variation of the energy at the touchdown (4.4) and the L_2 norm of the control input converge to zero as well. Those results show that the proposed algorithm generates an *optimal* passive gait by learning.

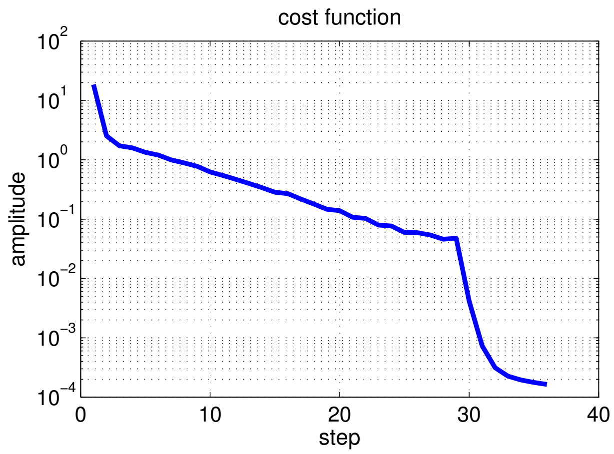

Figure 4.8: Cost function

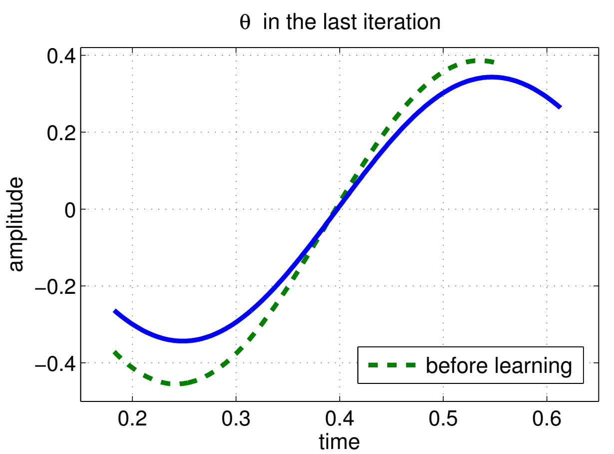

Figure 4.9: Response of θ

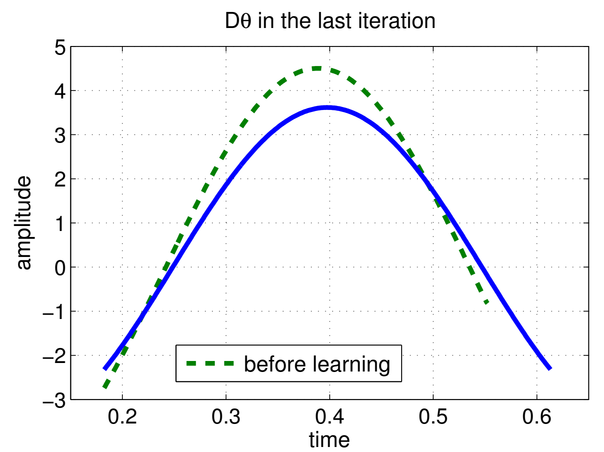

Figure 4.10: Response of $\dot{\theta}$

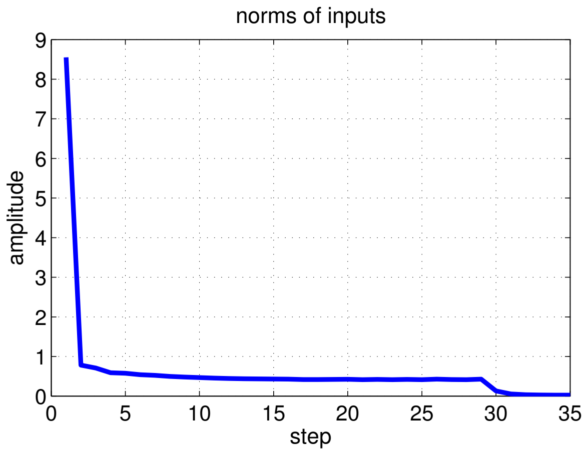

Figure 4.11: L_2 norm of the control inputs

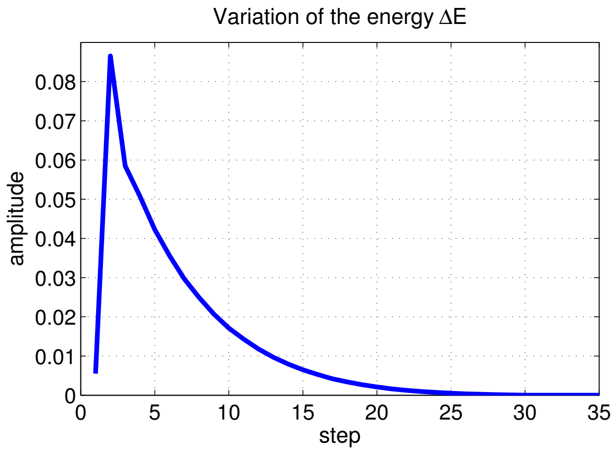

Figure 4.12: Variation of the energy at the touch down ΔE

Chapter 5

Optimal gait generation for the compass gait biped

So far we formulate the algorithm to generate optimal symmetric gait for the passive running robot and some numerical examples demonstrate the effectiveness of it. In this chapter, firstly, we investigate the applicability of the method to biped robots. We consider a simple planner biped robot called the *compass gait biped* in Section 5.1. Under an appropriate initial condition, this robot has a passive walking gait on a slope. We apply the proposed framework to the robot and try to generate an optimal symmetric gait which corresponds to a passive one in Section 5.2.

However, general walking gaits of legged robots including the compass gait biped are not symmetric unlike the passive running robot. Furthermore, in a walking motion, there are discrete state transitions caused by landing. Our proposed method in the previous chapter can not deal with them directly. Regarding this, secondly, we propose a framework to generate periodic trajectories which satisfy one of the necessary conditions for periodic gaits in Section 5.3. This technique estimates the state transition matrix by only using the information of the output and the angular velocities just before and after the transition. It implies that the proposed method does not require numerical transition models. Applying it to the compass gait biped, we can generate not only a passive walking gait on a slope but also an optimal gait on the level ground.

5.1 Description of the plant

Let us consider a full-actuated planar compass-like biped robot called the *compass gait biped* [7] depicted in Figure 5.1. The robot can walk down a gentle slope without any control inputs under appropriate initial conditions [10]. Table 5.1 shows physical parameters and variables. Furthermore, we assume some assumptions on this robot. Some important ones are as follows. The rest of them conforms [7].

Assumption 5.1 The foot does not bounce back nor slip on the ground (inelastic impulsive impact).

Assumption 5.2 Transfer of support between the stance and the swing legs is instantaneous.

We use number of notations with respect to the state. Table 5.2 summaries these ones. Here is a new input defined as $\bar{u} := (\bar{u}_1, \bar{u}_2)^{\mathrm{T}} = (u_1 + u_2, -u_2)^{\mathrm{T}}$ Then, the dynamics of this robot is

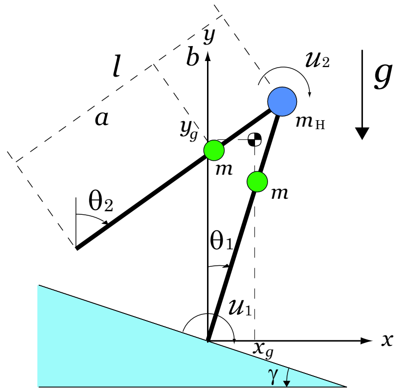

Figure 5.1: Model of the compass gait biped

Table 5.1: 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
$l = a + b$	total leg length	m
g	gravity acceleration	$\mathrm{m/s^2}$
γ	slope angle	rad
θ_1	stance leg angle w.r.t vertical	rad
θ_2	swing leg angle w.r.t vertical	rad
u_1	ankle torque	Nm
u_2	hip torque	Nm
x_g	horizontal position of C.o.M	m
y_g	vertical position of C.o.M	m

described by a Hamiltonian system in (2.1) with the Hamiltonian

$$H(\theta, \dot{\theta}, \bar{u}) = \frac{1}{2}\dot{\theta}^{\mathrm{T}} M(\theta)\dot{\theta} + V(\theta) - \bar{u}^{\mathrm{T}}\theta$$

where a positive definite matrix $M(\theta) \in \mathbb{R}^{2\times 2}$ denotes the inertia matrix and a scalar function $V(\theta) \in \mathbb{R}$ denotes the potential energy of the system. The matrices are as follows

$$M(\theta) = \begin{pmatrix} m_H l^2 + ma^2 + ml^2 & -mbl\cos(\theta_1 - \theta_2) \\ -mbl\cos(\theta_1 - \theta_2) & mb^2 \end{pmatrix}$$

$$V(\theta) = \{(m_H l + ma + ml)\cos\theta_1 - mb\cos\theta_2\}g. \tag{5.1}$$

Note that using Equation (5.1), the generalized momentum is described as $p = M(\theta)\dot{\theta}$.

Assumption 5.1 and Assumption 5.2 imply that there exists no double support phase. Following the law of conservation of the angular momentum, we can obtain a transition equation

Table 5.2: Some notations

Notation	Meaning
$q := (\theta_1, \theta_2)^{\mathrm{T}}$	generalized coordinate
$p := (p_1, p_2)^{\mathrm{T}}$	generalized momentum
$x := (q^{\mathrm{T}}, p^{\mathrm{T}})^{\mathrm{T}}$	state
$\theta := (\theta_1, \theta_2)^{\mathrm{T}}$	angles of legs
$\dot{\theta} := (\dot{\theta}_1, \dot{\theta}_2)^{\mathrm{T}}$	angular velocities of legs
$\Theta := (\theta^{\mathrm{T}}, \dot{\theta}^{\mathrm{T}})^{\mathrm{T}}$	angles and their velocities
$x^0 := ({q^0}^{\mathrm{T}}, {p^0}^{\mathrm{T}})^{\mathrm{T}}$	initial state
$x^1 := ({q^1}^{\mathrm{T}}, {p^1}^{\mathrm{T}})^{\mathrm{T}}$	terminal state
$\cdot^-$	just before transfer
$\cdot^+$	just after transfer
	Note that $\theta \equiv q$ and $x^- \equiv x^1$.

as

$$Q^+(\theta^-)\dot{\theta}^+ = Q^-(\theta^-)\dot{\theta}^-, \tag{5.2}$$

where $\dot{\theta}^-$ and $\dot{\theta}^+$ denote the angular velocities just before and just after the transition caused by a collision between a leg and the ground, respectively. Here let us note that $\theta^- \equiv \theta^+$. The matrices in (5.2) are as follows

$$Q^-(\theta^-) = \begin{pmatrix} (m_H l^2 + 2mal)\cos(\theta_1^- - \theta_2^-) - mab & -mab \\ -mab & 0 \end{pmatrix}$$

$$Q^+(\theta^-) = \begin{pmatrix} mb(b - l\cos(\theta_1^- - \theta_2^-)) & (ml^2 + ma^2 + m_H l^2) - mbl\cos(\theta_1^- - \theta_2^-) \\ mb^2 & -mbl\cos(\theta_1^- - \theta_2^-) \end{pmatrix}. \tag{5.3}$$

Here we rewrite the transition equation in (5.2) as

$$\dot{\theta}^+ = \mathbf{Q}(\theta^-)\dot{\theta}^-, \tag{5.4}$$

where $\mathbf{Q}(\theta^-) := {Q^+}^{-1}(\theta^-)\, Q^-(\theta^-)$.

5.2 Symmetric gait generation for the compass gait biped

In case of the passive running robot considered in Section 4.1, symmetry of the trajectory which holds the conditions (4.2) and (4.3) is a sufficient condition for a running gait. However, it does not hold for general walking robots. Let us investigate whether generating symmetric trajectories is reasonable for the compass gait biped. Phase portraits of typical passive walking gaits of the biped robot in [7] imply that these gaits are almost symmetric but not exactly. Here, we suppose optimal symmetric gaits generated by the algorithm in Section 4.3 exist in a neighborhood of passive walking ones. In this section, we apply the proposed framework in Section 4.3 to the compass gait biped depicted in Figure 5.1 and try to generate an optimal symmetric walking gait.

As Figure 5.2 illustrates, let us formulate the condition for symmetric gaits of the compass gait biped as follows

$$\theta(t-t^0)=-\theta(t^1-t) \tag{5.5}$$

$$\dot{\theta}(t-t^0)=\dot{\theta}(t^1-t). \tag{5.6}$$

Considering these conditions (5.5) and (5.6), The cost function is given by

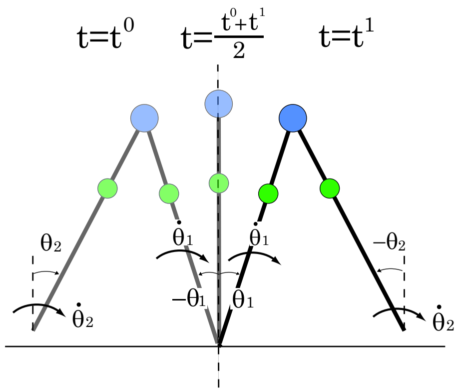

Figure 5.2: Illustration for a symmetric gait

$$\Gamma(\theta_j,\dot{\theta}_j,\bar{u}) := \frac{K_{\theta_j}}{2}\|\theta_j-(-\mathcal{R}(\theta_j))\|_{L_2}^2+\frac{K_{\dot{\theta}_j}}{2}\|\dot{\theta}_j-\mathcal{R}(\dot{\theta}_j)\|_{L_2}^2+\frac{K_{\bar{u}}}{2}\|\bar{u}\|_{L_2}^2 \quad (j=1,2). \tag{5.7}$$

Here K_{θ_j}, $K_{\dot{\theta}_j}$ $(j=1,2)$ and $K_{\bar{u}}$ represent appropriate positive gains. Then the iterative learning control law is given by

$$\bar{u}_{j(2i+1)}=\bar{u}_{j(2i)}+\epsilon_{(i)}\mathcal{R}\Big(2K_{\theta_j}(\mathrm{id}+\mathcal{R})(\theta_{j(2i)})-2K_{\dot{\theta}_j}(\mathrm{id}-\mathcal{R})(\ddot{\theta}_{j(2i)})\Big) \tag{5.8}$$

$$\bar{u}_{j(2i+2)}=\bar{u}_{j(2i)}-K_{j(2i)}\Big(K_{\bar{u}_j}\bar{u}_{j(2i)}+\frac{1}{\epsilon_{(i)}}\mathcal{R}(\theta_{j(2i+1)}-\theta_{j(2i)})\Big) \qquad (j=1,2). \tag{5.9}$$

Derivation of the law is omitted here due to the similar derivation process to (4.14) and (4.15). Regarding the dynamics of this robot, matrices J and R in (2.1) and T in (2.3) are obtained as

$$J=\begin{pmatrix}0 & I\\ -I & 0\end{pmatrix},\qquad R=\begin{pmatrix}0 & 0\\ 0 & 0\end{pmatrix},\qquad T=\begin{pmatrix}I & 0\\ 0 & -I\end{pmatrix}. \tag{5.10}$$

Using Equation (5.10), we can derive the update law for the initial state $x^0_{(i)}$ as the same manner in (4.18) that

$$q^0_{(i+1)}=q^0_{(i)}-K_{q^0(i)}p_{v(i)}\Big|_{t=t^1}$$
$$p^0_{(i+1)}=p^0_{(i)}-K_{p^0(i)}q_{v(i)}\Big|_{t=t^1}, \tag{5.11}$$

where $K_{q^0(\cdot)}$ and $K_{p^0(\cdot)}$ are appropriate positive matrices and q_v and p_v are the variations of q and p, respectively. Applying the learning procedure by (5.8), (5.9) and (5.11) to the compass gait biped, it is expected that we can generate an optimal symmetric gait close to a passive one. The validity of this method are examined by numerical simulations in Section 5.5.1.

5.3 Novel framework to generate periodic trajectories

In this section, we formulate a framework to generate periodic trajectories. Since these trajectories satisfy one of the necessary conditions for periodic gaits, this technique is expected to be valid for general walking robots. Furthermore, it can execute a learning procedure including discrete state transition without precise model of the transition mapping by estimating the transition matrix by only using the information of the output and the angular velocities just before and after the transition.

5.3.1 Definition of the cost function

This section defines a cost function which is expected to generate an optimal periodic trajectory. Now we propose a cost function as

$$\begin{aligned}\bar{\Gamma}(\Theta^0, \bar{u}, \Theta^1) &:= \frac{1}{2}(\Phi(\Theta^1) - \Theta^0)^{\mathrm{T}} \Lambda_x \, (\Phi(\Theta^1) - \Theta^0) + \frac{1}{2}\int_{t^0}^{t^1} \bar{u}(\tau)^{\mathrm{T}} \Lambda_{\bar{u}} \, \bar{u}(\tau) \mathrm{d}\tau \\ &= \frac{1}{2}(\psi_1(x^1) - \psi_0(x^0))^{\mathrm{T}} \Lambda_x \, (\psi_1(x^1) - \psi_0(x^0)) + \frac{1}{2}\int_{t^0}^{t^1} \bar{u}(\tau)^{\mathrm{T}} \Lambda_{\bar{u}} \, \bar{u}(\tau) \mathrm{d}\tau \\ &=: \Gamma(x^0, \bar{u}, x^1), \end{aligned} \tag{5.12}$$

where Λ_x and $\Lambda_{\bar{u}}$ represent appropriate positive definite matrices. Here $\Phi(\Theta^1)$ is defined to be the angles and their velocities just after the transition with exchanging legs as

$$\Phi(\Theta^1) := \begin{pmatrix} C\theta^+ \\ C\dot{\theta}^+ \end{pmatrix} = \begin{pmatrix} C & 0 \\ 0 & C\mathbf{Q}(\theta^-) \end{pmatrix} \Theta^1. \tag{5.13}$$

The matrix C exchanges the support and the swing leg angles as bellow

$$C := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{5.14}$$

Regarding the relation between x and Θ, we have

$$x = \begin{pmatrix} q \\ p \end{pmatrix} = \begin{pmatrix} \theta \\ M(\theta)\dot{\theta} \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & M(\theta) \end{pmatrix} \Theta. \tag{5.15}$$

Equations (5.13) and (5.15) imply that $\psi_1(x^1)$ and $\psi_0(x^0)$ are also defined as

$$\begin{aligned}\psi_1(x^1) &:= \begin{pmatrix} C & 0 \\ 0 & C\mathbf{Q}(\theta^-)M(\theta^-)^{-1} \end{pmatrix} x^1 \\ \psi_0(x^0) &:= \begin{pmatrix} I & 0 \\ 0 & M(\theta^0)^{-1} \end{pmatrix} x^0. \end{aligned} \tag{5.16}$$

Our previous cost functions (4.7) and (5.7) cannot take such state transitions caused by a collision between a leg and the ground into account, but this one in (5.12) can.

5.3.2 Derivation of iteration laws

This subsection derives the iteration law for the input and the initial conditions with respect to the cost function defined in (5.12) as in Section 4.3.

Here let us consider Equation (4.17) again.

$$\begin{aligned} \mathrm{d}\left(\Gamma((x^0, \bar{u}), \Sigma(x^0, \bar{u}))\right)(\mathrm{d}x^0, \mathrm{d}\bar{u}) = &\langle \Gamma'_{x^0} + \pi_{\mathbb{R}^n} \circ (\mathrm{d}\Sigma(x^0, \bar{u}))^*(\Gamma'_{x^1}, \Gamma'_y), \mathrm{d}x^0 \rangle \\ &+ \langle \Gamma'_{\bar{u}} + \pi_U \circ (\mathrm{d}\Sigma(x^0, \bar{u}))^*(\Gamma'_{x^1}, \Gamma'_y), \mathrm{d}\bar{u} \rangle, \end{aligned} \tag{5.17}$$

where

$$\Gamma'(x^0, \bar{u}, x^1, y) =: (\Gamma'_{x^0}, \Gamma'_{\bar{u}}, \Gamma'_{x^1}, \Gamma'_y)^{\mathrm{T}}. \tag{5.18}$$

Considering the cost function (5.12), (5.18) can be calculated as

$$\begin{aligned} \Gamma'_{x^0} &= -\frac{\mathrm{d}\psi_0(x^0)}{\mathrm{d}x^0}^{\mathrm{T}} \Lambda_x(\psi_1(x^1) - \psi_0(x^0)) \\ \Gamma'_{\bar{u}} &= \Lambda_{\bar{u}} \bar{u} \\ \Gamma'_{x^1} &= \frac{\mathrm{d}\psi_1(x^1)}{\mathrm{d}x^1}^{\mathrm{T}} \Lambda_x(\psi_1(x^1) - \psi_0(x^0)) \\ \Gamma'_y &= 0. \end{aligned} \tag{5.19}$$

Substitute Equation (5.19) for Equation (5.17) and suppose Equation (2.5) holds, then the steepest descent method implies that we should change the input $\bar{u}$ such that

$$\begin{aligned} \bar{u}_{(i+1)} &= \bar{u}_{(i)} - K_{(i)}\Big(\Gamma'_{\bar{u}(i)} + \pi_U \circ (\mathrm{d}\Sigma(x^0_{(i)}, \bar{u}_{(i)}))^*(\Gamma'_{x^1(i)}, \Gamma'_{y(i)})\Big) \\ &= \bar{u}_{(i)} - K_{(i)}\Big(\Gamma'_{\bar{u}(i)} + \mathcal{R} \circ \pi_Y \circ (\mathrm{d}\Sigma(x^0_{(i)}, \bar{u}_{(i)}))\Big(-(J-R)T\Gamma'_{x^1(i)}, \mathcal{R}(\Gamma'_{y(i)})\Big)\Big) \\ &\approx \bar{u}_{(i)} - K_{(i)}\Big\{\Gamma'_{\bar{u}(i)} + \frac{1}{\epsilon_{(i)}}\mathcal{R}\Big(\pi_Y \circ \Sigma\Big(x^0_{(i)} + \epsilon_{(i)}(-(J-R)T\Gamma'_{x^1(i)}), \bar{u}_{(i)} + \epsilon_{(i)}\mathcal{R}(\Gamma'_{y(i)})\Big) \\ &\qquad\qquad - \pi_Y \circ \Sigma(x^0_{(i)}, \bar{u}_{(i)})\Big)\Big\} \\ &= \bar{u}_{(i)} - K_{(i)}\Big\{\Lambda_{\bar{u}}\bar{u}_{(i)} + \frac{1}{\epsilon_{(i)}}\mathcal{R}\Big(\pi_Y \circ \Sigma\Big(x^0_{(i)} - \epsilon_{(i)}(J-R)T\frac{\mathrm{d}\psi_1}{\mathrm{d}x^1}^{\mathrm{T}}\Lambda_x(\psi_1(x^1_{(i)}) - \psi_0(x^0_{(i)})) \\ &\qquad\qquad , \bar{u}_{(i)}\Big) - \pi_Y \circ \Sigma(x^0_{(i)}, \bar{u}_{(i)})\Big)\Big\}, \end{aligned} \tag{5.20}$$

where i denotes the i-th iteration in laboratory experiment. Here the iterative learning control law without updating the initial conditions is given by

$$\begin{cases} x^0_{(2i+1)} = x^0_{(0)} - \epsilon_{(i)}(J-R)T\dfrac{\mathrm{d}\psi_1(x^1_{(2i)})}{\mathrm{d}x^1}^{\mathrm{T}}\Lambda_x(\psi_1(x^1_{(2i)}) - \psi_0(x^0_{(0)})) \\ \bar{u}_{(2i+1)} = \bar{u}_{(2i)} \end{cases} \tag{5.21}$$

$$\begin{cases} x^0_{(2i+2)} = x^0_{(0)} \\ \bar{u}_{(2i+2)} = \bar{u}_{(2i)} - K_{(2i)}\Big(\Lambda_{\bar{u}}\bar{u}_{(2i)} + \dfrac{1}{\epsilon_{(i)}}\mathcal{R}(y_{(2i+1)} - y_{(2i)})\Big) \end{cases} \tag{5.22}$$

provided that the initial input $u_{(0)}$ is equivalent to zero and the first initial condition $x^0_{(0)}$ is appropriately chosen. Here $\epsilon_{(\cdot)}$ denotes a sufficiently small positive constant.

Then let us derive the updating law for the initial conditions in the same manner as in Section 4.3. For Equation (5.17), The initial conditions should be updated as

$$\mathrm{d}x^0 = -K_{x^0}\Big(\Gamma'_{x^0} + \pi_{\mathbb{R}^n} \circ (\mathrm{d}\Sigma(x^0, \bar{u}))^*(\Gamma'_{x^1}, \Gamma'_y)\Big) \tag{5.23}$$

where K_{x^0} is an appropriate positive gain.

Let us substitute Equation (5.19) for Equation (5.23). Then we calculate the initial state of $(\mathrm{d}\Sigma(x^0, u))^*$ with respect to the input $(\Gamma'_{x^1}, \Gamma'_y)$, that is, $x^0_a := \pi_{\mathbb{R}^n} \circ (\mathrm{d}\Sigma(x^0, u))^*(\Gamma'_{x^1}, \Gamma'_y)$. As mentioned in Chapter 2, the initial state is calculated as $x^0_a = -T^{-1}(J-R)^{-1}\bar{x}_v(t^0)$. The state of the dynamics in (2.4) is identified with the time-reversal version of that in (2.3), so we can write the initial state of the adjoint of the variational system as $x^0_a = -T^{-1}(J-R)^{-1}x_v(t^1)$ under certain circumstances as explained in Remark 2.2. For the dynamics of the compass gait biped depicted in Figure 5.1, $x^0_a := ({q^0_a}^{\mathrm{T}}, {p^0_a}^{\mathrm{T}})^{\mathrm{T}}$ is calculated as follows

$$\begin{pmatrix} q^0_a \\ p^0_a \end{pmatrix} = -T^{-1}(J-R)^{-1}\begin{pmatrix} q_v(t^1) \\ p_v(t^1) \end{pmatrix} = \begin{pmatrix} p_v(t^1) \\ q_v(t^1) \end{pmatrix}, \tag{5.24}$$

where $(q_v(t^1)^{\mathrm{T}}\ p_v(t^1)^{\mathrm{T}})^{\mathrm{T}} =: x_v(t^1)$ and

$$T = \begin{pmatrix} I & 0 \\ 0 & -I \end{pmatrix}, \ J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}, \ R = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}. \tag{5.25}$$

Using Equations (5.23) and (5.24), we should update the initial conditions as

$$\begin{aligned} x^0_{(2i+2)} &= x^0_{(2i)} - K_{x^0(2i)}\left(-\frac{\mathrm{d}\psi_0(x^0_{(2i)})}{\mathrm{d}x^0}^{\mathrm{T}}\Lambda_x(\psi_1(x^1_{(2i)}) - \psi_0(x^0_{(2i)})) + \begin{pmatrix} p_v(t^1) \\ q_v(t^1) \end{pmatrix}\right) \\ &= x^0_{(2i)} - K_{x^0(2i)}\left(-\frac{\mathrm{d}\psi_0(x^0_{(2i)})}{\mathrm{d}x^0}^{\mathrm{T}}\Lambda_x(\psi_1(x^1_{(2i)}) - \psi_0(x^0_{(2i)})) + \begin{pmatrix} p^1_{(2i+1)} - p^1_{(2i)} \\ q^1_{(2i+1)} - q^1_{(2i)} \end{pmatrix}\right), \end{aligned} \tag{5.26}$$

where $K_{x^0(\cdot)}$ represents an appropriate positive gain.

Combining Equations (5.21), (5.22) and (5.26), we can derive the iteration law as

$$\begin{cases} x^0_{(2i+1)} = x^0_{(2i)} - K_{x^0(2i-2)}\Bigg(-\dfrac{\mathrm{d}\psi_0(x^0_{(2i-2)})}{\mathrm{d}x^0}^{\mathrm{T}} \Lambda_x\big(\psi_1(x^1_{(2i-2)}) - \psi_0(x^0_{(2i-2)})\big) \\ \qquad\qquad + \begin{pmatrix} p^1_{(2i-1)} - p^1_{(2i-2)} \\ q^1_{(2i-1)} - q^1_{(2i-2)} \end{pmatrix}\Bigg) - \epsilon_{(i)}(J-R)T\dfrac{\mathrm{d}\psi_1(x^1_{(2i)})}{\mathrm{d}x^1}^{\mathrm{T}} \Lambda_x(\psi_1(x^1_{(2i)}) - \psi_0(x^0_{(2i)})) \\ \bar{u}_{(2i+1)} = \bar{u}_{(2i)} \end{cases} \tag{5.27}$$

$$\begin{cases} x^0_{(2i+2)} = x^0_{(2i)} - K_{x^0(2i)}\Bigg(-\dfrac{\mathrm{d}\psi_0(x^0_{(2i)})}{\mathrm{d}x^0}^{\mathrm{T}} \Lambda_x\big(\psi_1(x^1_{(2i)}) - \psi_0(x^0_{(2i)})\big) + \begin{pmatrix} p^1_{(2i+1)} - p^1_{(2i)} \\ q^1_{(2i+1)} - q^1_{(2i)} \end{pmatrix}\Bigg) \\ \bar{u}_{(2i+2)} = \bar{u}_{(2i)} - K_{(2i)}\Big(\Lambda_{\bar{u}}\bar{u}_{(2i)} + \dfrac{1}{\epsilon_{(i)}}\mathcal{R}(y_{(2i+1)} - y_{(2i)})\Big) \end{cases} . \tag{5.28}$$

Equations (5.27) and (5.28) imply that the procedure requires the precise knowledge of $\psi_1(x^1)$, $\psi_0(x^0)$, $\mathrm{d}\psi_1(x^1)/\mathrm{d}x^1$ and $\mathrm{d}\psi_0(x^0)/\mathrm{d}x^0$ which include the transition equation and the inertia matrix. Regarding real robots, however, the numerical model of the state transition does not hold exactly. For example, the angular momentum is not always preserved. So in the next section, we propose a learning algorithm without using this information.

5.4 Modification of the iteration law

In this section, we reconsider the iteration law in (5.27) and (5.28) in order not to use the knowledge of the state transition mapping and the inertia matrix, i.e., $\psi_1(x^1)$, $\psi_0(x^0)$, $\mathrm{d}\psi_1(x^1)/\mathrm{d}x^1$ and $\mathrm{d}\psi_0(x^0)/\mathrm{d}x^0$ by estimating the transition mapping. This method allows one to execute the learning procedure by only using the information of the output and the angular velocities just before and after transition.

5.4.1 Input iteration law

We consider the input iteration law in (5.21) and (5.22). The updating law for the initial conditions is considered later.

Equations (5.1) and (5.3) imply that $M(\theta)$ and $\mathbf{Q}(\theta)$ depend only on $\theta_1 - \theta_2$. Let us define the following variable α that

$$\theta_1 - \theta_2 =: \alpha \in \mathbb{S}. \tag{5.29}$$

Firstly, let us calculate the gradient of the cost function (5.12) with respect to Θ^1. Here we obtain

$$\frac{\mathrm{d}\Phi(\Theta^1)}{\mathrm{d}\Theta^1}^{\mathrm{T}} \Lambda_x(\Phi(\Theta^1) - \Theta^0) =: \bar{\Gamma}'_{\Theta^1}. \tag{5.30}$$

However, the knowledge of the transition mapping is also required to calculate $\mathrm{d}\Phi(\Theta^1)/\mathrm{d}\Theta^1$ as in the case of $\mathrm{d}\psi_1(x^1)/\mathrm{d}x^1$. Here we define the difference $\Delta\Phi_i \ \ (i=1,\cdots,4)$ as

$$\Delta\Phi_i := \Phi(\Theta^1+\Delta\Theta^1_i)-\Phi(\Theta^1) = \frac{\mathrm{d}\Phi(\Theta^1)}{\mathrm{d}\Theta^1}\Delta\Theta^1_i + o_{4\times 1}(\|\epsilon_\Theta\|) \qquad (i=1,\cdots,4),$$

where $\Delta\Theta^1_i := (\Delta\theta_i^{-\mathrm{T}}, \Delta\dot{\theta}_i^{-\mathrm{T}})^{\mathrm{T}}$ represents a perturbation of Θ^1 which holds $\|\Delta\Theta^1_{(\cdot)}\| < \epsilon_\Theta \ll 1$ and $o_{m\times n}(\cdot)$ denotes an $m\times n$ matrix consisted of $o(\cdot)$ elements . Then we have

$$(\Delta\Phi_1,\Delta\Phi_2,\Delta\Phi_3,\Delta\Phi_4) = \frac{\mathrm{d}\Phi(\Theta^1)}{\mathrm{d}\Theta^1}(\Delta\Theta^1_1,\Delta\Theta^1_2,\Delta\Theta^1_3,\Delta\Theta^1_4) + o_{4\times 4}(\|\epsilon_\Theta\|).$$

Suppose that $\Delta\Theta^1_i \ \ (i=1,\cdots,4)$ is linear independent with each other , we can calculate $\mathrm{d}\Phi(\Theta^1)/\mathrm{d}\Theta^1$ as

$$\frac{\mathrm{d}\Phi(\Theta^1)}{\mathrm{d}\Theta^1} \approx (\Delta\Phi_1,\Delta\Phi_2,\Delta\Phi_3,\Delta\Phi_4)(\Delta\Theta^1_1,\Delta\Theta^1_2,\Delta\Theta^1_3,\Delta\Theta^1_4)^{-1}. \tag{5.31}$$

Furthermore, we suppose the following assumptions.

Assumption 5.3 The following condition holds in a neighborhood of the origin.

$$\left\|\frac{\mathrm{d}M}{\mathrm{d}\alpha}\right\|\epsilon_\Theta \ll \|M\|$$

Assumption 5.4 The following condition holds in a neighborhood of the origin.

$$\left\|\frac{\mathrm{d}\mathbf{Q}}{\mathrm{d}\alpha}\right\|\epsilon_\Theta \ll \|\mathbf{Q}\|$$

Using (5.29) and Assumption 5.4, the matrix $\mathbf{Q}$ is calculated as

$$\begin{aligned}\mathbf{Q}(\alpha^-+\Delta\alpha_i^-) &= \mathbf{Q}(\alpha^-) + \frac{\mathrm{d}\mathbf{Q}(\alpha^-)}{\mathrm{d}\alpha}\Delta\alpha_i^- + o(|\epsilon_\Theta|)\\ &\approx \mathbf{Q}(\alpha^-).\end{aligned} \tag{5.32}$$

Equations (5.13) and (5.32) imply that $\Delta\Phi_i$ is given by

$$\begin{aligned}\Delta\Phi_i &= \Phi(\Theta^1+\Delta\Theta^1_i)-\Phi(\Theta^1)\\ &= \begin{pmatrix} C & 0\\ 0 & C\mathbf{Q}(\theta^-+\Delta\theta_i^-)\end{pmatrix}\begin{pmatrix}\theta^-+\Delta\theta_i^-\\ \dot{\theta}^-+\Delta\dot{\theta}_i^-\end{pmatrix} - \begin{pmatrix} C & 0\\ 0 & C\mathbf{Q}(\theta^-)\end{pmatrix}\begin{pmatrix}\theta^-\\ \dot{\theta}^-\end{pmatrix}\\ &\approx \begin{pmatrix} C & 0\\ 0 & C\end{pmatrix}\begin{pmatrix}\Delta\theta_i^+\\ \Delta\dot{\theta}_i^+\end{pmatrix}.\end{aligned} \tag{5.33}$$

Using Equations (5.31) and (5.33), we can calculate that

$$\begin{aligned}\frac{\mathrm{d}\Phi(\Theta^1)}{\mathrm{d}\Theta^1} \approx &\begin{pmatrix} C & 0\\ 0 & C\end{pmatrix}\left[\begin{pmatrix}\Delta\theta_1^+\\ \Delta\dot{\theta}_1^+\end{pmatrix}\begin{pmatrix}\Delta\theta_2^+\\ \Delta\dot{\theta}_2^+\end{pmatrix}\begin{pmatrix}\Delta\theta_3^+\\ \Delta\dot{\theta}_3^+\end{pmatrix}\begin{pmatrix}\Delta\theta_4^+\\ \Delta\dot{\theta}_4^+\end{pmatrix}\right]\\ &\times\left[\begin{pmatrix}\Delta\theta_1^-\\ \Delta\dot{\theta}_1^-\end{pmatrix}\begin{pmatrix}\Delta\theta_2^-\\ \Delta\dot{\theta}_2^-\end{pmatrix}\begin{pmatrix}\Delta\theta_3^-\\ \Delta\dot{\theta}_3^-\end{pmatrix}\begin{pmatrix}\Delta\theta_4^-\\ \Delta\dot{\theta}_4^-\end{pmatrix}\right]^{-1}.\end{aligned} \tag{5.34}$$

Secondly, let us consider $\Phi(\Theta^1) - \Theta^0$. By Equations (5.4) and (5.13), we can calculate that

$$\Phi(\Theta^1) - \Theta^0 = \begin{pmatrix} C & 0 \\ 0 & C\mathbf{Q}(\theta^-) \end{pmatrix} \begin{pmatrix} \theta^- \\ \dot{\theta}^- \end{pmatrix} - \begin{pmatrix} \theta^0 \\ \dot{\theta}^0 \end{pmatrix}$$

$$= \begin{pmatrix} C & 0 \\ 0 & C \end{pmatrix} \begin{pmatrix} \theta^+ \\ \dot{\theta}^+ \end{pmatrix} - \begin{pmatrix} \theta^0 \\ \dot{\theta}^0 \end{pmatrix}. \tag{5.35}$$

Finally, using Equations (5.30), (5.34) and (5.35), the variation with respect to Θ is given by

$$-(J-R)T\,\bar{\Gamma}'_{\Theta^1} \approx -(J-R)T \left[\begin{pmatrix} \Delta\theta_1^- \\ \Delta\dot{\theta}_1^- \end{pmatrix} \begin{pmatrix} \Delta\theta_2^- \\ \Delta\dot{\theta}_2^- \end{pmatrix} \begin{pmatrix} \Delta\theta_3^- \\ \Delta\dot{\theta}_3^- \end{pmatrix} \begin{pmatrix} \Delta\theta_4^- \\ \Delta\dot{\theta}_4^- \end{pmatrix} \right]^{-\mathrm{T}} \tag{5.36}$$

$$\times \left[\begin{pmatrix} \Delta\theta_1^+ \\ \Delta\dot{\theta}_1^+ \end{pmatrix} \begin{pmatrix} \Delta\theta_2^+ \\ \Delta\dot{\theta}_2^+ \end{pmatrix} \begin{pmatrix} \Delta\theta_3^+ \\ \Delta\dot{\theta}_3^+ \end{pmatrix} \begin{pmatrix} \Delta\theta_4^+ \\ \Delta\dot{\theta}_4^+ \end{pmatrix} \right]^{\mathrm{T}} \begin{pmatrix} C & 0 \\ 0 & C \end{pmatrix} \Lambda_x \left(\begin{pmatrix} C & 0 \\ 0 & C \end{pmatrix} \begin{pmatrix} \theta^+ \\ \dot{\theta}^+ \end{pmatrix} - \begin{pmatrix} \theta^0 \\ \dot{\theta}^0 \end{pmatrix} \right).$$

Here let us note that

$$-(J-R)T = \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix}$$

by Equation (5.25). Using Equation (5.15), Equation (5.21) becomes

$$\begin{cases} \begin{pmatrix} I & 0 \\ 0 & M(\theta^0_{(2i+1)}) \end{pmatrix} \begin{pmatrix} \theta^0_{(2i+1)} \\ \dot{\theta}^0_{(2i+1)} \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & M(\theta^0_{(0)}) \end{pmatrix} \begin{pmatrix} \theta^0_{(0)} \\ \dot{\theta}^0_{(0)} \end{pmatrix} + \epsilon_{(i)} \begin{pmatrix} I & 0 \\ 0 & M(\theta^0_{(0)}) \end{pmatrix} \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix} \bar{\Gamma}'_{\Theta^1} \\ \bar{u}_{(2i+1)} = \bar{u}_{(2i)} \end{cases} \tag{5.37}$$

Since $\epsilon_{(\cdot)}$ is a design parameter, we can set $\epsilon_{(\cdot)}$ so that $\|\theta^0_{(\cdot)} - \theta^0_{(0)}\| < \epsilon_\Theta$ holds. Then Assumption 5.3 implies

$$M(\theta^0_{(\cdot)}) \approx M(\theta^0_{(0)}). \tag{5.38}$$

Consequently, the iteration law without updating the initial conditions (5.21) and (5.22) can be calculated as

$$\begin{cases} \begin{pmatrix} \theta^0_{(2i+1)} \\ \dot{\theta}^0_{(2i+1)} \end{pmatrix} = \begin{pmatrix} \theta^0_{(0)} \\ \dot{\theta}^0_{(0)} \end{pmatrix} + \epsilon_{(i)} \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix} \left[\begin{pmatrix} \Delta\theta^-_{1(2i)} \\ \Delta\dot{\theta}^-_{1(2i)} \end{pmatrix} \begin{pmatrix} \Delta\theta^-_{2(2i)} \\ \Delta\dot{\theta}^-_{2(2i)} \end{pmatrix} \begin{pmatrix} \Delta\theta^-_{3(2i)} \\ \Delta\dot{\theta}^-_{3(2i)} \end{pmatrix} \begin{pmatrix} \Delta\theta^-_{4(2i)} \\ \Delta\dot{\theta}^-_{4(2i)} \end{pmatrix} \right]^{-\mathrm{T}} \\ \qquad \times \left[\begin{pmatrix} \Delta\theta^+_{1(2i)} \\ \Delta\dot{\theta}^+_{1(2i)} \end{pmatrix} \begin{pmatrix} \Delta\theta^+_{2(2i)} \\ \Delta\dot{\theta}^+_{2(2i)} \end{pmatrix} \begin{pmatrix} \Delta\theta^+_{3(2i)} \\ \Delta\dot{\theta}^+_{3(2i)} \end{pmatrix} \begin{pmatrix} \Delta\theta^+_{4(2i)} \\ \Delta\dot{\theta}^+_{4(2i)} \end{pmatrix} \right]^{\mathrm{T}} \begin{pmatrix} C & 0 \\ 0 & C \end{pmatrix} \\ \qquad \times \Lambda_x \left(\begin{pmatrix} C & 0 \\ 0 & C \end{pmatrix} \begin{pmatrix} \theta^+_{(2i)} \\ \dot{\theta}^+_{(2i)} \end{pmatrix} - \begin{pmatrix} \theta^0_{(0)} \\ \dot{\theta}^0_{(0)} \end{pmatrix} \right) \\ \bar{u}_{(2i+1)} = \bar{u}_{(2i)} \end{cases} \tag{5.39}$$

$$\begin{cases} \begin{pmatrix} \theta^0_{(2i+2)} \\ \dot{\theta}^0_{(2i+2)} \end{pmatrix} = \begin{pmatrix} \theta^0_{(0)} \\ \dot{\theta}^0_{(0)} \end{pmatrix} \\ \\ \bar{u}_{(2i+2)} = \bar{u}_{(2i)} - K_{(2i)}\Big(\Lambda_{\bar{u}}\bar{u}_{(2i)} + \dfrac{1}{\epsilon_{(i)}}\mathcal{R}(y_{(2i+1)} - y_{(2i)})\Big) \end{cases}. \tag{5.40}$$

The input iteration law given by (5.39) and (5.40) does not require the knowledge of the state transition equation nor the inertia matrix. We can calculate this by only using the information of the output and the angular velocities just before and just after the transition. Four pairs of difference data, i.e. $\Delta(\cdot)$, in (5.39) can be calculated by using data in the previous experiments. For only the first experiment, five prior experimental data is necessary.

5.4.2 Updating law for the initial conditions

Let us consider the updating law for the initial conditions in (5.26) again.

$$x^0_{(2i+2)} = x^0_{(2i)} - K_{x^0(2i)}\left(\Gamma'_{x^0(2i)} + \begin{pmatrix} p^1_{(2i+1)} - p^1_{(2i)} \\ q^1_{(2i+1)} - q^1_{(2i)} \end{pmatrix}\right) \tag{5.41}$$

The gradient of the cost function (5.12) with respect to Θ^0 is calculated as

$$\begin{aligned} \bar{\Gamma}'_{\Theta^0} &:= -\Lambda_x(\Phi(\Theta^1) - \Theta^0) \\ &= -\Lambda_x\left(\begin{pmatrix} C & 0 \\ 0 & C \end{pmatrix}\begin{pmatrix} \theta^+ \\ \dot{\theta}^+ \end{pmatrix} - \begin{pmatrix} \theta^0 \\ \dot{\theta}^0 \end{pmatrix}\right). \end{aligned}$$

In the same manner as mentioned in the previous subsection, Equation (5.41) is calculated that

$$\begin{aligned} &\begin{pmatrix} I & 0 \\ 0 & M(\theta^0_{(2i+2)}) \end{pmatrix}\begin{pmatrix} \theta^0_{(2i+2)} \\ \dot{\theta}^0_{(2i+2)} \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & M(\theta^0_{(2i)}) \end{pmatrix}\begin{pmatrix} \theta^0_{(2i)} \\ \dot{\theta}^0_{(2i)} \end{pmatrix} - K_{x^0(2i)} \\ &\quad \times\left[-\begin{pmatrix} I & 0 \\ 0 & M(\theta^0_{(2i)}) \end{pmatrix}\Lambda_x\left(\begin{pmatrix} C & 0 \\ 0 & C \end{pmatrix}\begin{pmatrix} \theta^1_{(2i)} \\ \dot{\theta}^{1+}_{(2i)} \end{pmatrix} - \begin{pmatrix} \theta^0_{(2i)} \\ \dot{\theta}^0_{(2i)} \end{pmatrix}\right)\right. \\ &\quad \left.+\left(\begin{pmatrix} M(\theta^1_{(2i+1)}) & 0 \\ 0 & I \end{pmatrix}\begin{pmatrix} \dot{\theta}^1_{(2i+1)} \\ \theta^1_{(2i+1)} \end{pmatrix} - \begin{pmatrix} M(\theta^1_{(2i)}) & 0 \\ 0 & I \end{pmatrix}\begin{pmatrix} \dot{\theta}^1_{(2i)} \\ \theta^1_{(2i)} \end{pmatrix}\right)\right]. \end{aligned} \tag{5.42}$$

Here we suppose the following assumption.

Assumption 5.5 For any integer i, if $\|\Theta^0_{(2i+1)} - \Theta^0_{(2i)}\| \ll 1$ holds, then $\|\Theta^1_{(2i+1)} - \Theta^1_{(2i)}\| \ll 1$.

As is mentioned in the previous subsection, we can set a design parameter ϵ so that $\|\Theta^0_{(2i+1)} - \Theta^0_{(2i)}\| \ll 1$ holds. For example, the target system satisfies the Lipschitz condition, Assumption 5.5 holds.

Using Equation (5.38) and Assumption 5.5, consequently, we obtain

$$\begin{pmatrix} \theta^0_{(2i+2)} \\ \dot{\theta}^0_{(2i+2)} \end{pmatrix} = \begin{pmatrix} \theta^0_{(2i)} \\ \dot{\theta}^0_{(2i)} \end{pmatrix} - K_{x^0(2i)} \Bigg[- \Lambda_x \left(\begin{pmatrix} C & 0 \\ 0 & C \end{pmatrix} \begin{pmatrix} \theta^1_{(2i)} \\ \dot{\theta}^{1+}_{(2i)} \end{pmatrix} - \begin{pmatrix} \theta^0_{(2i)} \\ \dot{\theta}^0_{(2i)} \end{pmatrix} \right) + \bar{M}_{(2i)} \begin{pmatrix} \dot{\theta}^1_{(2i+1)} - \dot{\theta}^1_{(2i)} \\ \theta^1_{(2i+1)} - \theta^1_{(2i)} \end{pmatrix} \Bigg], \tag{5.43}$$

where $\bar{M}_{(\cdot)}$ represents an appropriate positive matrix.

5.5 Numerical examples

We apply the proposed algorithms to the compass gait biped. Firstly as mentioned in Section 5.2, we generate an optimal symmetric gait minimizing the L_2 norm of the control input, which is almost close to a passive walking gait but not exactly. Secondly, we generate an exact passive walking gait applying our novel framework to generate an optimal periodic trajectories as mentioned in Section 5.3. Finally, we also generate an optimal walking gait on the level ground by the proposed framework.

The concrete values of parameters of the robot are $m_H = 10$ [kg], $m = 5$ [kg], $l = 1$ [m], $a = b = 0.5$ [m].

5.5.1 Symmetric gait

We apply the iteration laws (5.8), (5.9) and (5.11) proposed in Section 5.2 to the compass gait biped to generate optimal symmetric gait. We set the small constant parameter $\delta = 0.05$ to determine the convergence of the L_2 norms of the control inputs. The initial condition is

$$x^0_{(1)} = (\theta^0_{1(1)}, \theta^0_{2(1)}, p^0_{1(1)}, p^0_{2(1)}) = (-0.25, 0.35, 18.4683, -1.8510),$$

and we set the gain parameters as follows: $K_{\theta_1} = 0.5$, $K_{\theta_2} = 0.5$, $K_{\dot{\theta}_1} = 1 \times 10^{-5}$, $K_{\dot{\theta}_2} = 1 \times 10^{-5}$, $K_{\bar{u}} = 1.5$,

$$K_{q^0} = \begin{pmatrix} 0.08 & 0 \\ 0 & 0.5 \end{pmatrix}, \quad K_{p^0} = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.5 \end{pmatrix},$$

and the slope angle $\gamma = 0.05$ [rad]. Figure 5.6 shows the procedure terminates at the 76th Step which means we execute 152 simulations.

Figure 5.3 shows the history of the cost function (5.7) along the iteration. It monotonically decreases, which implies that the output trajectory converges to the *optimal* one consequently at each learning experiment. Figure 5.4 shows responses of θ_1 and $\dot{\theta}_1$ and Figure 5.5 shows responses of θ_2 and $\dot{\theta}_2$, respectively. These signals at the last step in the proposed method in solid lines and those of the initial trajectory in dotted lines, respectively. Figures 5.4, 5.5 and 5.6 imply that we can generate optimal gait which almost satisfy the condition (5.6) and the L_2 norm of the control input converge to sufficiently small. Figure 5.7 shows that θ_1-$\dot{\theta}_1$ phase portrait of the generated gait. It is close to that of the typical passive walking gait. We also observe 200 steps consecutive walking on the slope. Those results show that the proposed algorithm generates optimal symmetric gait close to passive walking gait.

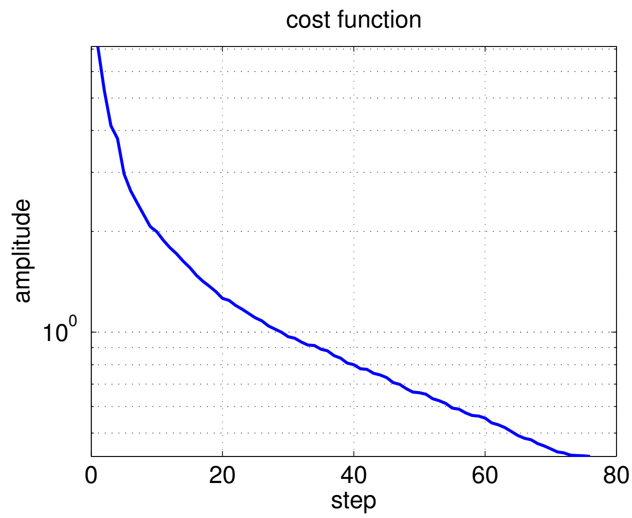

Figure 5.3: Cost function

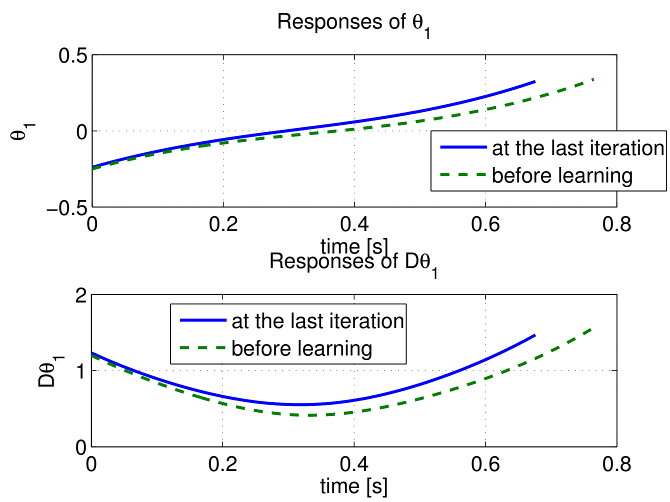

Figure 5.4: Responses of θ_1 and $\dot{\theta}_1$

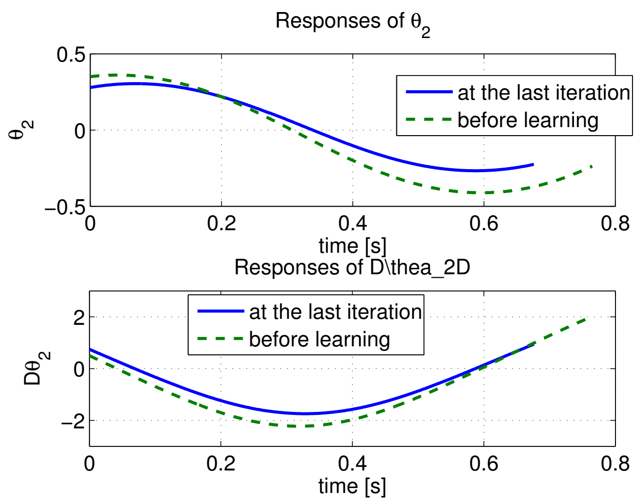

Figure 5.5: Responses of θ_2 and $\dot{\theta}_2$

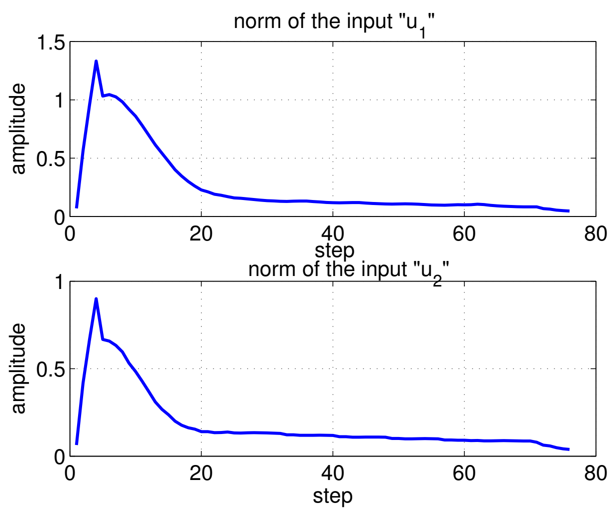

Figure 5.6: L_2 norm of control inputs u_1 and u_2

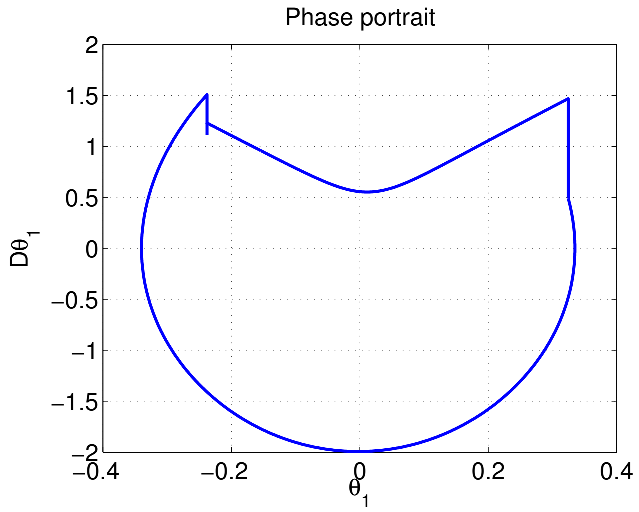

Figure 5.7: Phase portrait

5.5.2 Passive walking gait

In the previous example, we can not generate an exact passive walking gait because this gait is not exactly symmetric. So, in this example we apply the proposed algorithm in Section 5.3 to the compass gait biped. We set the small constant $\delta = 0.02$. We use the initial condition and the slope angle as the same as those in the previous example. We execute 200 simulations with the gain parameters $\Lambda_x = \mathrm{diag}(0.01, 0.01, 0.08, 0.16)$, $\Lambda_{\bar{u}} = 1 \times 10^{-5}$, $K_{x^0} = \mathrm{diag}(7 \times 10^{-4}, 2.5 \times 10^{-3}, 0.5, 0.1)$.

Figure 5.8 shows the history of the cost function (5.12) along the iteration. decreasing monotonically, which implies that the output trajectory converges to the *optimal* one smoothly at each learning experiment. Figures 5.9 and 5.10 show responses of the pairs of θ_1 and $\dot{\theta}_1$ and θ_2 and $\dot{\theta}_2$ at the last step in solid lines and those of the initial trajectory in dotted lines, respectively. Figure 5.11 shows that θ_1-$\dot{\theta}_1$ phase portrait of the generated gait. It is the same as that of the typical passive walking gait. We also observe 200 steps consecutive walking on the slope with zero input under the generated initial condition. Those results show that the proposed algorithm generates an *optimal* passive gait by learning.

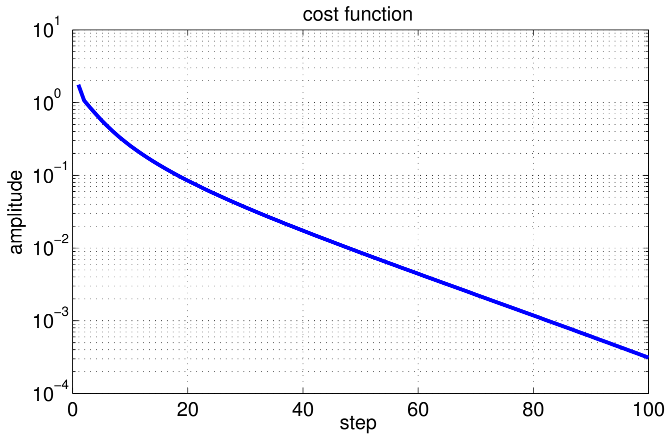

Figure 5.8: Cost function

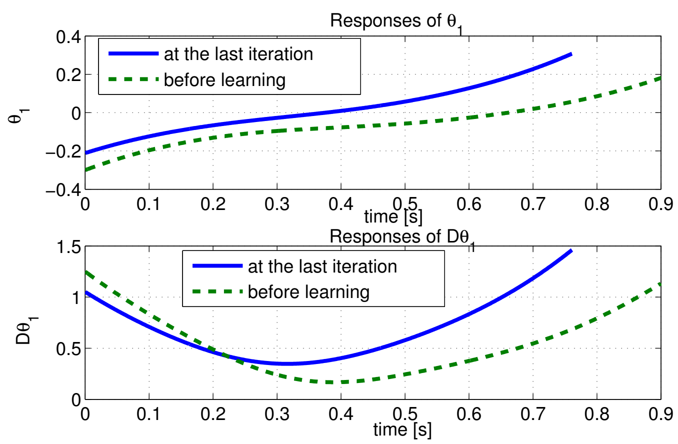

Figure 5.9: Responses of θ_1 and $\dot{\theta}_1$

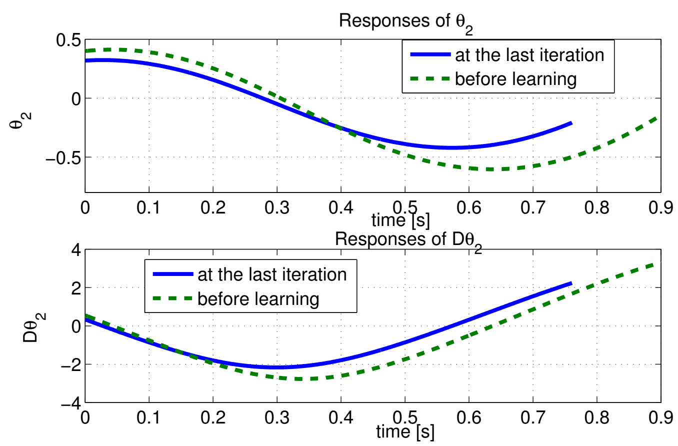

Figure 5.10: Responses of θ_2 and $\dot{\theta}_2$

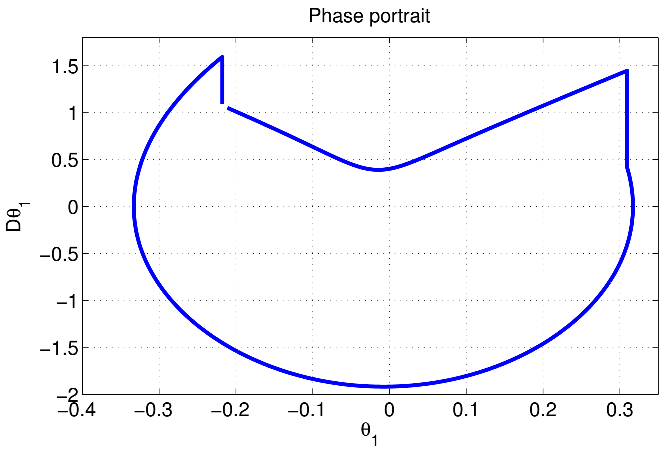

Figure 5.11: Phase portrait

5.5.3 Optimal gait on the level ground

In this example, we apply the iteration laws (5.39), (5.40) and (5.43) to generate optimal gait on the level ground without using the knowledge of the discrete state transition mapping nor the inertia matrix.

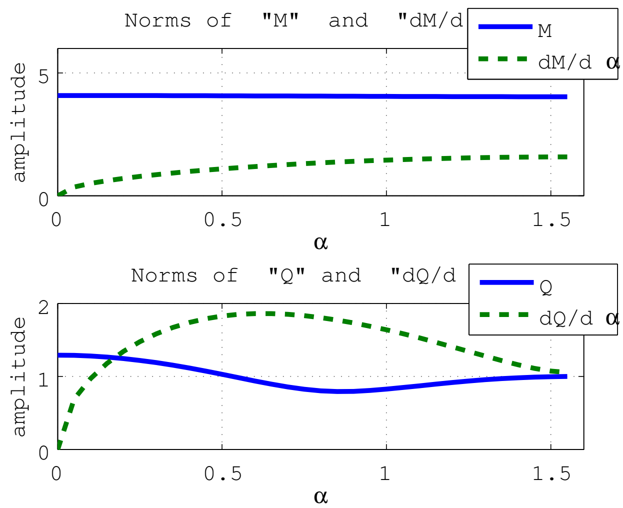

Figure 5.12: Comparison of norms

Figure 5.12 shows the comparison of $\|M\|$ and $\left\|\dfrac{\mathrm{d}M}{\mathrm{d}\alpha}\right\|$ and $\|\mathbf{Q}\|$ and $\left\|\dfrac{\mathrm{d}\mathbf{Q}}{\mathrm{d}\alpha}\right\|$, respectively. It implies that Assumption 5.3 and Assumption 5.4 hold considering the compass gait biped.

We choose the initial condition as

$$x^0_{(1)} = (\theta^0_{1(1)}, \theta^0_{2(1)}, p^0_{1(1)}, p^0_{2(1)}) = (-0.3, 0.3, 18.4683, -1.8510),$$

and we set the design parameters as follows:

$$\Lambda_x = \begin{pmatrix} 2 & 0 & 0 & 0 \\ a0 & 2.5 & 0 & 0 \\ 0 & 0 & 1\times 10^{-4} & 0 \\ 0 & 0 & 0 & 1\times 10^{-14} \end{pmatrix}, \quad \Lambda_{\bar{u}} = 1\times 10^{-5},$$

$\epsilon_{(\cdot)} = 2\times 10^{-3}$ and $K_{(\cdot)} = \mathrm{diag}(0.01, 0.01)$. We execute 300 simulations.

Figure 5.13 shows the history of the cost function (5.12) along the iteration. It monotonically decreases, which implies that the output trajectory converges to an *optimal* one smoothly. Figure 5.14 shows that θ_1-$\dot{\theta}_1$ phase portrait of the generated gait. It exhibits that a stable limit cycle is generated. Figure 5.15 shows an optimal control inputs generated in the last iteration. Furthermore, We observe 200 steps consecutive walking with these inputs. Those results show that the proposed algorithm generates an *optimal* gait on the level ground.

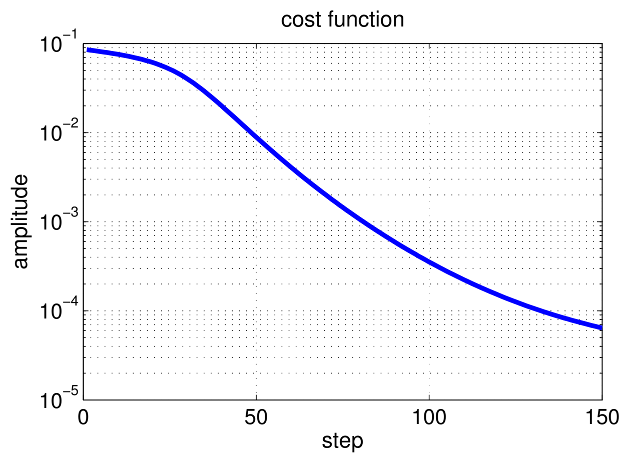

Figure 5.13: Cost function

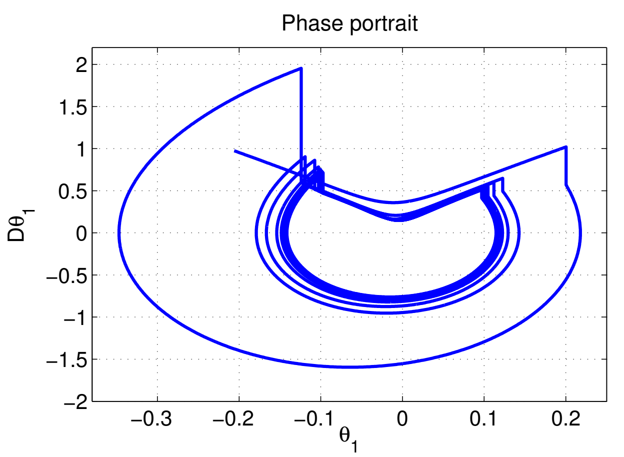

Figure 5.14: Phase portrait

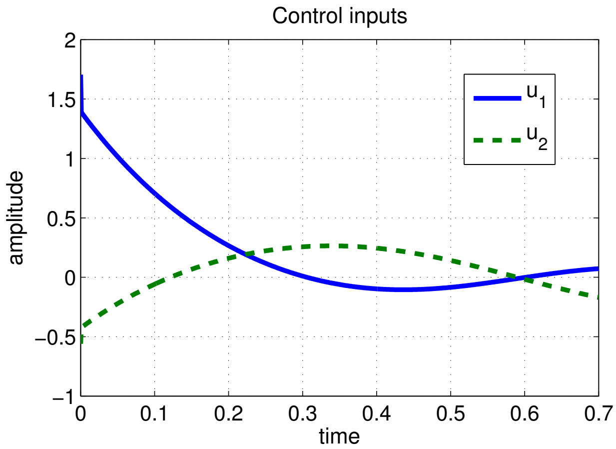

Figure 5.15: Control inputs u_1 and u_2 in the last iteration

Chapter 6

Conclusion

In this paper, we have proposed a novel framework to generate optimal gait for legged robots via iterative learning control based on variational symmetry. Firstly, we define a cost function to achieve time symmetric trajectories minimizing the L_2 norm of the control input. However, the conventional iterative learning control method can not deal with a cost function which is a functional of $\dot{y}$ the time derivative of the output y. For conventional mechanical systems, the signal $\dot{y}$ represents the velocity, which plays an important role. To solve this problem, we propose an extension of iterative learning control by employing a pseudo adjoint of the time derivative operator. This method allows one to take the time derivative of the output into account. By this technique, we can generate a sub-optimal symmetric gait minimizing the L_2 norm of the control input of a one-legged running robot called a passive running robot. Though this robot can run with zero control input under appropriate initial conditions, the proposed method can not generate optimal gait with zero input. This is because the algorithm can not take into account the variation of the initial condition.

Secondly, we propose an algorithm to generate optimal gait trajectories by adopting a novel update law for the initial conditions as well as that for the feedforward input. This framework allows one to generate optimal symmetric gait of the passive running robot in a sense that the corresponding input converges to zero. Then we investigate the applicability of the method to a simple planner biped robot called the compass gait biped. Symmetry of the trajectory is a sufficient condition for running gaits of the passive running robot. However, it does not hold for general walking robots including the compass gait biped. Since phase portraits of typical passive walking gaits of the biped robot are almost symmetric but not exactly, we suppose optimal symmetric gaits generated by our method exist in a neighborhood of passive walking gaits. Some numerical examples have shown that we can generate optimal symmetric walking gaits close to passive walking ones.

Thirdly, we propose a novel framework to generate periodic trajectories which satisfy one of the necessary conditions for periodic gaits. In a walking motion, there are discrete sate transitions caused by landing. This novel technique can achieve a learning procedure including such a discrete state transition without any precise model of the transition mapping. In numerical walking analysis, the state transition mapping is often derived by the conservation law of angular momentum. However, this law does not hold exactly in considering real robots. We apply this method to the compass gait biped to generate a passive walking gait on a slope and ,furthermore, an optimal gait on the level ground.

Consequently, the proposed learning algorithm can solve a class of optimal control problems without precise knowledge of the plant system nor the discrete state transition mapping. It is expected that it will be applicable to optimal gait generation problems for more complicated walking robots.

Bibliography

[1] M. Ahmadi and M. Buehler. Stable control of a simulated one-legged running robot with hip and leg compliance. *IEEE Trans. Robotics and Automation*, 13(1):96–104, 1997.

[2] S. Arimoto, S. Kawamura, and F. Miyazaki. Bettering operation of robotics. *Journal of Robotic Systems*, 1(2):123–140, 1984.

[3] 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*, 20(3):565–573, 2004.

[4] K. Fujimoto, T. Horiuchi, and T. Sugie. Optimal control of Hamiltonian systems with input constraints via iterative learning. In *Proc. 42nd IEEE Conf. on Decision and Control*, pages 4387–4392, 2003.

[5] K. Fujimoto and T. Sugie. Iterative learning control of Hamiltonian systems: I/O based optimal control approach. *IEEE Trans. Autom. Contr.*, 48(10):1756–1761, 2003.

[6] A. Goswami, B. Espiau, and A. Keramane. Limit cycles in a passive compass gait biped and passivity-mimicking control laws. *Autonomous Robots*, 4(3):273–286, 1997.

[7] A. Goswami, B. Thuilot, and B. Espiau. Compass-like biped robot part i: Stability and bifurcation of passive gaits. *INRIA Research Report*, (2996), 1996.

[8] S. Hyon. Hamiltonian-based running control of dynamic legged robots. *Systems, Control and Information*, 49(7):260–265, 2005. (in Japanese).

[9] S. Hyon and T. Emura. Energy-preserving control of passive one-legged running robot. *Advanced Robotics*, 18(4):357–381, 2004.

[10] T. McGeer. Passive dynamic walking. *Int. J. Robotics Research*, 9(2):62–82, 1990.

[11] K. Osuka and K. Kirihara. Motion analysis and experiments of passive walking robot quartet ii. In *Proc. the 2000 IEEE Int. Conf. on Robotics and Automation*, pages 3052–3056, 2000.

[12] A. Sano, Y. Ikemata, and H. Fujimoto. Analysis of dynamics of passive walking from storage energy and supply rate. In *Proc. the 2003 IEEE Int. Conf. on Robotics and Automation*, pages 2478–2483, 2003.

[13] S. Satoh, K. Fujimoto, and S. Hyon. Gait generation for a hopping robot via iterative learning control based on variational symmetry. In *Proc. IFAC 3rd Workshop on Lagrangian and Hamiltonian Methods for Nonlinear Control*, pages 125–130, 2006.

[14] S. Satoh, K. Fujimoto, and S. Hyon. Optimal gait generation for the compass gait biped based on variational symmetry. In *Proc. 35th SICE Symposium on Control Theory (in Japanese)*, pages 357–360, 2006.

[15] M. W. Spong. Passivity-base control of the compass gait biped. In *Proc. of IFAC World Congress*, pages 19–23, 1999.

[16] C. Thompson and M. Raibert. Passive dynamic running. In V. Hayward and O. khatib, editors, *Experimental Robotics I*, volume 139 of *Lecture Notes in Control and Information Science*, pages 74–83. Springer-Verlag, Berlin, 1989.