Gait generation via unified learning optimal control of Hamiltonian systems

Satoshi Satoh†,*, Kenji Fujimoto‡ and Sang-Ho Hyon§

†Division of Mechanical Systems and Applied Mechanics, Faculty of Engineering, Hiroshima University, 1-4-1, Kagamiyama, Higashi-Hiroshima 739-8527, Japan
‡Department of Aerospace Engineering, Graduate School of Engineering, Kyoto University, Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501, Japan
§Department of Robotics, Ritsumeikan University, Noji Higashi 1-1-1, Kusatsu, Shiga 525-8577, Japan

SUMMARY

This paper proposes a repetitive control type optimal gait generation framework by executing learning control and parameter tuning. We propose a learning optimal control method of Hamiltonian systems unifying iterative learning control (ILC) and iterative feedback tuning (IFT). It allows one to simultaneously obtain an optimal feedforward input and tuning parameter for a plant system, which minimizes a given cost function. In the proposed method, a virtual constraint by a potential energy prevents a biped robot from falling. The strength of the constraint is automatically mitigated by the IFT part of the proposed method, according to the progress of trajectory learning by the ILC part.

KEYWORDS: Gait generation; Biped robots; Repetitive control; Iterative learning control; Hamiltonian systems.

1. Introduction

Recently, control of walking robots has become an active research area. As the technology for walking robots evolves, an optimization problem of gaits with respect to energy consumption becomes increasingly important. However, it is difficult to design *a priori* the optimal walking trajectory. Most of walking pattern generation and control methods have been based on the zero moment point (ZMP) criterion.[1–4] This method can generate a stable walking pattern of a walking robot with multiple degrees of freedom (DoF), which means that the robot does not fall, and its implementation is relatively easy. However, the trajectory based on the ZMP is designed heuristically in many cases and energy efficiency has not been considered sufficiently. Passive dynamic walker[5–8] also attracts attention. This robot has a certain simple structure and it walks down on a gentle slope with no actuation but gravity. Walking control methods based on it have been proposed by many researchers, see e.g. refs. [9–11]. Although the generated gaits are energy efficient, these methods are only applicable to certain specially structured robots so far. Besides, walking control methods using virtual constraints based on the output zeroing control are proposed in refs. [12, 13]. Since appropriately chosen holonomic constraints reduce the order of the system, these methods are applicable to the robots with multiple DoF, under actuation and so on. However, there is no systematic way to choose constraints achieving a target motion, and such constraints often consume a large amount of control energy. Gait generation methods based on central pattern generator[14] or statistical learning[15–17] are also studied. The first method[14] considers a humanoid robot and constructs a good controller for it, where a leg trajectory is first generated by a network of oscillators and then each joint trajectory is calculated via inverse kinematics. However, the number of necessary oscillators and how to couple them are heuristic. The second method[15] optimizes a nominal trajectory via differential dynamic programming (DDP), which is a second-order local trajectory optimization method. Although the authors empirically show that the generated gait overcomes certain disturbances by combining DDP and a minimax criterion, a nominal trajectory is required and the state and the cost function are discretized. Since the first two methods are model-based learning, the plant models are necessary. On the contrary, the latter two methods are model free and both are based on the policy gradient reinforcement learning. The third method[16] considers a simple 3D biped robot and generates an optimal control policy which makes the robot robust against small slope changing. However, it requires information of a nominal periodic trajectory and it cannot work with high dimensional problems due to the so-called curse of dimensionality. The last method[17] generalizes the path integral stochastic optimal control approach.[18] Although this approach generally requires the plant model, this method provides a special control structure for the rigid-body dynamics such that a desired trajectory is parameterized as a policy and the control policy is optimized. However, this method requires the knowledge of the desired trajectory dynamics and it is only applicable to the special control structure.

On the contrary, we have considered that physical properties and learning control are useful tools to tackle this challenging problem. So far, in refs. [19–21], we have proposed an optimal gait generation method by modifying iterative learning control (ILC) based on a symmetric property called variational symmetry of Hamiltonian systems.[22] Since this technique is a motivation of the proposed method in this paper, whose details are mentioned

* Corresponding author. E-mail: s.satoh@ieee.org

in the next paragraph, we first elaborate on it. Hamiltonian systems[23,24] have been introduced to represent physical systems and they explicitly possess good properties for the control design such as passivity, symmetry and so on. We consider biped robots as Hamiltonian systems, rather than as just nonlinear systems, and take advantage of their physical properties. ILC based on variational symmetry of Hamiltonian systems[22] allows one to solve a class of optimal control problems by iteration of laboratory experiments. Thanks to the symmetric property, it does not require the precise knowledge of the plant model and it can directly deal with infinite-dimensional optimal control problems without any finite-dimensional approximation. This is a big difference with the conventional ILC, e.g. ref. [25], in that the conventional ILC is only applicable to trajectory tracking control problems. However, although the ILC method in ref. [22] works well for some control problems, there are mainly two difficulties to apply it to the optimal gait generation problem. The first one is that this method deals with a functional of the input and the output as a cost function, but it cannot take the time derivative of the output into account. Since this signal represents the generalized velocity of mechanical systems, its behavior severely affects the walking motion. The other difficulty is that it cannot take discontinuous state transitions into account. Such transitions involved in general walking motions also have to be considered. In order to solve these problems for the optimal gait generation, we have modified the above ILC method in refs. [19–21]. First, we proposed an extension by employing a pseudo-adjoint of the time derivative operator. This method enables one to deal with a functional of the time derivative of the output as a cost function. Second, we considered a state transition mapping of the collision to be a general nonlinear function with respect to the state just before touchdown, and proposed an estimation method of the mapping by the least squares with stored experimental data.

Since our previous method in refs. [19–21] is classified as ILC framework, it requires that we repeat laboratory experiments under the same initial condition as well as many conventional results, e.g. refs. [22, 25]. However, this initialization procedure is sometimes strict, because it is difficult to realize the desired initial velocity of the mechanical systems including walking robots. To solve the problem, this paper proposes a new repetitive control type optimal gait generation framework by executing ILC and a parameter tuning. Here, we refer to iterative feedback tuning (IFT) based on variational symmetry in ref. [26] as the parameter tuning method of Hamiltonian systems. We newly propose a learning optimal control method of Hamiltonian systems by unifying ILC and IFT. While ILC[22,25] is to find an optimal feedforward input minimizing a given cost function, IFT[26–28] is to find optimal parameters of a given feedback controller. A conventional repetitive control method[29] is also a kind of a learning method for a trajectory tracking control problem with time periodic reference trajectories. Since an iteration procedure of the proposed framework is automatically executed and eventually an optimal periodic gait is expected to be generated, it is classified as repetitive control rather than ILC. The idea of the proposed method is as follows. First, we add a constraint by adding a virtual

potential energy to prevent the robot from falling. Second, we execute the modified learning procedure in refs. [19–21] mentioned in the previous paragraph. The virtual potential energy is designed so that the motion of the robot is restricted to a symmetric trajectory. It saves one resetting the initial conditions. Third, by regarding the potential gain for the constraint as a tuning parameter, we execute parameter tuning to mitigate the strength of the virtual constraint automatically according to the progress of learning control. Consequently, it is expected to generate an optimal gait without constraint eventually. In this method, ILC and IFT of Hamiltonian systems are utilized simultaneously. However, since both methods influence each other, they regularly cannot be used simultaneously. In order to take interference of both methods into account, we introduce an extended system which again has variational symmetry. The extended system instead of the original plant system enables one to apply ILC and IFT simultaneously. The proposed learning optimal control allows one to simultaneously obtain an optimal feedforward input and tuning parameter for a plant system, which minimizes a cost function. We summarize the advantages of the proposed method compared with the other (particularly learning) methods:

- it can generate an optimal trajectory as a solution to an infinite-dimensional optimal control problem without any discretization of the state space nor the time horizon nor parameterization of the policy;
- since it can directly provide the gradient of the cost function, we do not have to search the whole space;
- it does not require time-consuming calculations, such as large-scale inverse matrices and calculations of dynamics, nor an enormous amount of memory; and
- it does not require the plant model nor any information of nominal/desired trajectory.

This paper grew out of our previous reports.[30,31] In ref. [30], the interference of ILC and IFT was not handled well, and advanced considerations are added to ref. [31]. The remainder of the paper is organized as follows. In Section 2, ILC and IFT methods based on variational symmetry of Hamiltonian systems[22,26] are referred to. In Section 3, we introduce a compass-like biped robot with torso as a plant system. In Section 4, we equip a constraint via a virtual potential energy and propose a unified learning optimal control method of ILC and IFT. Then, we summarize the proposed optimal gait generation method. In Section 5, some numerical simulations demonstrate the effectiveness of the proposed method. In Sections 6 and 7, discussion and conclusions are given.

2. Variational Symmetry of Hamiltonian Systems and Its Application to ILC and IFT

This section considers a Hamiltonian system and its symmetric property called variational symmetry.[22] This property relates the variational system of the Hamiltonian system and its adjoint one. Then, utilizing this property the ILC[22] and IFT[26] methods are briefly referred to. These methods play important roles in the proposed framework.

2.1. Hamiltonian systems and variational symmetry

We consider a Hamiltonian system with dissipation[23,24] $\Sigma^{x_{t^0}} : U \to Y : u \mapsto y$ as

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

This system is one of the representations of the physical systems and it includes not only the conventional Hamiltonian systems but also passive electro-mechanical systems, mechanical systems with nonholonomic constraints and so on. Here, $x(t) \in X$, $u \in U$ and $y \in Y$ with Hilbert spaces X, U and Y describe the state, the input and the output, respectively. Typically, $X = \mathbb{R}^n$ and $U, Y = L_2^m[t^0, t^1]$ on a finite time interval $[t^0, t^1]$. The structure matrix $J(x) \in \mathbb{R}^{n \times n}$ and the dissipation matrix $R(x) \in \mathbb{R}^{n \times n}$ are skew symmetric and symmetric positive semi-definite, respectively. The variational system $\delta\Sigma^{x_{t^0}}$ is the Fréchet derivative of the system $\Sigma^{x_{t^0}}$. The definition of the Fréchet derivative is as follows.

Definition 1. *Consider an operator $f : \tilde{\Xi} \to \Psi$ with Banach spaces Ξ and Ψ, and an open subset $\tilde{\Xi} \subset \Xi$. f is said to be Fréchet differentiable at $x \in \tilde{\Xi}$ if there exists an operator $\delta f : \tilde{\Xi} \times \Xi \to \Psi$ such that $\delta f(x)(\xi)$ is linear in ξ and the following holds: For any $\xi \in \Xi$ such that $x + \xi \in \tilde{\Xi}$,*

$$f(x + \xi) = f(x) + \delta f(x)(\xi) + o(\|\xi\|_\Xi), \quad (2)$$

where

$$\lim_{\|\xi\|_\Xi \to 0} \frac{o(\|\xi\|_\Xi)}{\|\xi\|_\Xi} = 0.$$

Under these circumstances, $\delta f(x)(\cdot)$ is called the Fréchet derivative of f at x.

The calculation of the Fréchet derivative is also utilized for derivation of the learning iteration law, e.g. Eqs. (11), (29) and (41). Here, let us refer to the following lemma. It relates the variational system to its adjoint one, which appears in solving optimal control problems.

Lemma 1 (ref. [22]). *Consider the Hamiltonian system $\Sigma^{x_{t^0}}$ in (1). Suppose that J and R are constant and that there exists a nonsingular matrix $T \in \mathbb{R}^{n \times n}$ satisfying*

$$J = -TJT^{-1}, \qquad R = TRT^{-1}, \quad (3)$$

$$\frac{\partial^2 H(x, u)}{\partial(x, u)^2} = \begin{pmatrix} T & O_{nn} \\ O_{nn} & I_n \end{pmatrix} \frac{\partial^2 H(x, u)}{\partial(x, u)^2} \begin{pmatrix} T^{-1} & O_{nn} \\ O_{nn} & I_n \end{pmatrix}. \quad (4)$$

Here, I_i and O_{ij} represent $i \times i$ identity matrix and $i \times j$ zero matrix, respectively. Then the variational system of $\Sigma^{x_{t^0}}$ is described by another linear Hamiltonian system $y_v = (\delta\Sigma^{x_{t^0}}(u))(u_v)$:

$$\begin{cases} \dot{x} = (J - R)\dfrac{\partial H(x, u)^\top}{\partial x}, & x(t^0) = x_{t^0}, \\ \dot{x}_v = (J - R)\dfrac{\partial H_v(x, u, x_v, u_v)^\top}{\partial x_v}, \\ y_v = -\dfrac{\partial H_v(x, u, x_v, u_v)^\top}{\partial u_v}, \end{cases} \quad (5)$$

with the initial state $x_v(t^0) = 0$. Here the controlled Hamiltonian $H_v(x, u, x_v, u_v)$ is given by

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

Furthermore, a state-space realization of the adjoint system with zero terminal state, denoted by $y_a = (\delta\Sigma^{x_{t^0}}(u))^*(u_a)$, coincides with a time-reversal version of that of the variational system with zero initial state (5). This property is called variational symmetry *of Hamiltonian systems.*

Remark 1. *In ref. [22], the variational system with nonzero initial state and its adjoint are also considered.*

Regarding variational symmetry, the following theorem is useful.

Theorem 1 (ref. [26]). *Consider the Hamiltonian system (1) and suppose that conditions of Lemma 1 are satisfied. Suppose moreover that, for two inputs $v, w \in U$, the corresponding state trajectories $\phi(t), \psi(t) \in X, t \in [t^0, t^1]$ satisfy*

$$\mathcal{R}\left(\frac{\partial^2 H(x, u)}{\partial(x, u)^2}\bigg|_{\substack{x = \phi \\ u = v}}\right) = \frac{\partial^2 H(x, u)}{\partial(x, u)^2}\bigg|_{\substack{x = \psi \\ u = w}}, \quad (6)$$

where $\mathcal{R}$ represents the time reversal operator on $[t^0, t^1]$ defined by

$$\mathcal{R}(u)(t) = u(t^1 - t + t^0), \quad \forall t \in [t^0, t^1]. \quad (7)$$

Then, variational symmetry leads to the following relationship:

$$(\delta\Sigma^{\phi(t^0)}(v))^* = \mathcal{R}(\delta\Sigma^{\psi(t^0)}(w))\mathcal{R}. \quad (8)$$

The above operator maps from $L_2^m[t^0, t^1]$ to $L_2^m[t^0, t^1]$.

Remark 2. *A state trajectory under which the configuration coordinate q and the phase coordinate $\dot{q}$ satisfy*

$$\begin{aligned} q(t) &= q(t^1 - t + t^0), \\ \dot{q}(t) &= -\dot{q}(t^1 - t + t^0), \quad \forall t \in [t^0, t^1], \end{aligned} \quad (9)$$

represents a time-symmetric motion with respect to the middle point of time $t = (t^0 + t^1)/2$. We call the trajectory satisfying the condition (9) symmetric trajectory. Suppose a state trajectory ϕ corresponding to an input v is symmetric

trajectory. Then, the condition (6) *in Theorem 1 is satisfied with $\psi = \phi$ and $w = v$, and therefore the following simpler relationship than Eq.* (8) *holds:*

$$(\delta \Sigma^{\phi(t^0)}(v))^* = \mathcal{R}(\delta \Sigma^{\phi(t^0)}(v))\mathcal{R}. \tag{10}$$

From the relations (8) and (10), we convert the adjoint system to the corresponding variational one, and calculate it by a difference approximation. Then, the input–output mapping of the adjoint system can be obtained by only using the input–output data of the original system, see Eqs. (13) and (31). This is a key technique of our learning framework based on variational symmetry.

2.2. ILC and IFT based on variational symmetry

We review some results of ILC in ref. [22] and IFT in ref. [26]. They have a common feature that they take advantage of variational symmetry of Hamiltonian systems. The objective of ILC is to find an optimal feedforward input which minimizes a given cost function, while that of IFT is to find optimal parameters of a given feedback controller.

First, let us refer to ILC. Consider the system $\Sigma^{x_t 0} : U \to Y$ in (1) and a cost function $\hat{\Gamma}(u, y) : U \times Y \to \mathbb{R}$. By utilizing the relation $y = \Sigma^{x_t 0}(u)$, the cost function can be written by $\Gamma(u) : U \to \mathbb{R} := \hat{\Gamma}(u, \Sigma^{x_t 0}(u))$. Let us calculate the Fréchet derivative (Definition 1) of the cost function in order to obtain the gradient with respect to the input u as follows:

$$\delta \Gamma(u)(\delta u) = \langle \nabla_u \hat{\Gamma}(u, y), \delta u \rangle_U + \langle \nabla_y \hat{\Gamma}(u, y), \delta y \rangle_Y$$
$$= \langle \underbrace{\nabla_u \hat{\Gamma}(u, y) + (\delta \Sigma^{x_t 0}(u))^*(\nabla_y \hat{\Gamma}(u, y))}_{=: \nabla \Gamma(u)}, \delta u \rangle_U. \tag{11}$$

The well-known Riesz representation theorem and the linearity of the Fréchet derivative guarantee that there exist functions $\nabla_u \hat{\Gamma}(u, y)$ and $\nabla_y \hat{\Gamma}(u, y)$ as above. Since $\nabla \Gamma(u)$ in Eq. (11) represents the gradient of the cost function with respect to u, the steepest descent method implies that one can reduce the cost function down at least to a local minimum by the following iteration law with a positive definite matrix $K_{(i)}$:

$$u_{(i+1)} = u_{(i)} - K_{(i)} \nabla \Gamma(u_{(i)}). \tag{12}$$

Here, the subscript (i) denotes the ith iteration in a laboratory experiment. However, calculation of the gradient $\nabla \Gamma(u)$ generally requires the precise knowledge of the plant system $\Sigma^{x_t 0}$, because it contains the output signal of the adjoint system $(\delta \Sigma^{x_t 0}(u))^*$ corresponding to the input signal $\nabla_y \hat{\Gamma}(u, y)$. If the assumption in Theorem 1 holds, the following approximation is obtained from Eq. (8) with a sufficiently small constant $\epsilon > 0$:

$$(\delta \Sigma^{x_t 0}(u))^*(\nabla_y \hat{\Gamma}(u, y))$$
$$= \mathcal{R}(\delta \Sigma^{\psi(t^0)}(w))(\mathcal{R}(\nabla_y \hat{\Gamma}(u, y))),$$
$$\approx \frac{\Sigma^{\psi(t^0)}(w + \epsilon \mathcal{R}(\nabla_y \hat{\Gamma}(u, y))) - \Sigma^{\psi(t^0)}(w)}{\epsilon}. \tag{13}$$

The approximation (13) enables one to execute the iteration procedure with Eq. (12) by only using the input–output data of the plant system $\Sigma^{x_t 0}$.

Second, let us refer to IFT. Here, we consider a feedback system of a Hamiltonian system with a generalized canonical transformation[32] so that the feedback system is also described by another Hamiltonian system in the form of (1). Therefore, the system parameters of the closed-loop system H_c, J_c and R_c generally depend on the parameters of the feedback controller to be adjusted. For simplicity, in this paper, it is supposed that only the Hamiltonian function H_c depends on the tuning parameter $\rho \in \mathbb{R}^s$. The case where J_c and R_c also depend on ρ is considered in ref. [26]. Consider a feedback system of the form (1) with a Hamiltonian $H_c(x, u, \rho)$. In the IFT method proposed in ref. [26], the tuning parameter is considered to be a virtual input for the Hamiltonian system, and a corresponding output is induced so that the input–output map has variational symmetry. Let us introduce the following zeroth-order hold operator $\mathfrak{h}$, which maps the parameter $\rho \in \mathbb{R}^s$ to $u_\rho \in L_2^s[t^0, t^1]$ in order to define a virtual input:

$$\mathfrak{h} : \mathbb{R}^s \to L_2^s[t^0, t^1] : u_\rho(t) := (\mathfrak{h}(\rho))(t) \equiv \rho, \forall t \in [t^0, t^1]. \tag{14}$$

For the virtual input u_ρ, let us consider the following input–output map $y_\rho = \Sigma_\rho^{x_t 0, u}(u_\rho)$:

$$\begin{cases} \dot{x} = (J - R) \dfrac{\partial H_c(x, u, u_\rho)^\top}{\partial x}, \ x(t^0) = x_{t^0}, \\ y_\rho = -\dfrac{\partial H_c(x, u, u_\rho)^\top}{\partial u_\rho}, \end{cases} \tag{15}$$

Since this map $\Sigma_\rho^{x_t 0, u}$ is a Hamiltonian system of the form (1), Lemma 1 and Theorem 1 imply that it has variational symmetry with some conditions. Here the following property with respect to $\mathfrak{h}$ defined in Eq. (14) is exhibited, which is utilized in the IFT algorithm.

Lemma 2 (ref. [26]). $\mathfrak{h}^*$ *is characterized by the following equation for any $\xi \in L_2^s[t^0, t^1]$:*

$$\mathfrak{h}^*(\xi) = \int_{t^0}^{t^1} \xi(t) \mathrm{d}t. \tag{16}$$

By utilizing the input–output map (15) and Eq. (16), an iteration algorithm for IFT can be derived in the manner similar to the case of ILC. Roughly speaking, consider the system (15) and a cost function $\hat{\Gamma}_\rho(u_\rho, y_\rho) : L_2^s \times L_2^s \to \mathbb{R}$. By replacing u, y, $\Sigma^{x_t 0}$ and $\hat{\Gamma}$ in Eq. (11) with u_ρ, y_ρ, $\Sigma_\rho^{x_t 0, u}$ and $\hat{\Gamma}_\rho$, we have $\delta \Gamma_\rho(u_\rho)(\delta u_\rho) = \langle \nabla \Gamma_\rho(u_\rho), \delta u_\rho \rangle$. From the linearity of the Fréchet derivative and the operator $\mathfrak{h}$, that is,

$$\delta \mathfrak{h}(\rho) \mathrm{d}\rho = \mathfrak{h}(\rho + \mathrm{d}\rho) - \mathfrak{h}(\rho) + o(\|\mathrm{d}\rho\|)$$
$$= \mathfrak{h}(\mathrm{d}\rho) + o(\|\mathrm{d}\rho\|)$$

Table I. Parameters and variables.

Notation	Meaning	Unit
m_T	Torso mass	kg
m_L	Leg mass	kg
a	Length from m_L to the ground	m
b	Length from the hip to m_L	m
$l = a + b$	Total leg length	m
c	Length from the hip to m_T	m
g	Gravity acceleration	m/s^2
q^1	Stance leg angle with respect to vertical	rad
q^2	Swing leg angle with respect to vertical	rad
q^3	Torso angle with respect to vertical	rad
v^1	Ankle torque	Nm
v^2	Torque relatively applied from torso to swing leg	Nm
v^3	Torque relatively applied from torso to stance leg	Nm
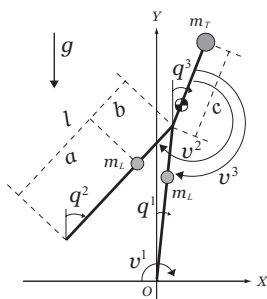
Fig. 1. Model of the compass gait biped with a torso.

and from Definition 1, we have

$$\delta u_\rho = \delta \mathfrak{h}(\rho) d\rho = \mathfrak{h}(d\rho). \qquad (17)$$

Then, $\delta \Gamma_\rho(u_\rho)(\delta u_\rho)$ reduces to

$$\langle \nabla \Gamma_\rho(u_\rho), \delta u_\rho \rangle = \langle \mathfrak{h}^*(\nabla \Gamma_\rho(u_\rho)), d\rho \rangle. \qquad (18)$$

Since $\mathfrak{h}^*(\nabla \Gamma_\rho(u_\rho))$ in Eq. (18) represents the gradient of the cost function with respect to the tuning parameter ρ, the iteration law for IFT is given by with a positive definite matrix $K_{\rho(i)}$ (for calculation of $\mathfrak{h}^*$, see Eq. (16)):

$$\rho_{(i+1)} = \rho_{(i)} - K_{\rho(i)} \int_{t^0}^{t^1} \nabla \Gamma_\rho(u_{\rho(i)}(t)) dt. \qquad (19)$$

3. Description of the plant

Let us consider a full-actuated planar compass-like biped robot with a torso depicted in Fig. 1. The legs without knees and the torso are rigid bars, and they are connected by a frictionless hinge at the hip. A 1-period of walking describes the period between the take-off of one foot from the ground and its subsequent landing. Table I shows physical parameters and variables. In this paper, we define the input u as

$$u = (u^1, u^2, u^3)^\top := (v^1 - v^3, -v^2, v^2 + v^3)^\top \qquad (20)$$

Table II. Some notations.

Notation	Meaning
$q := (q^1, q^2, q^3)^\top$	Angles of legs and a torso
$\dot{q} := (\dot{q}^1, \dot{q}^2, \dot{q}^3)^\top$	Angular velocities of legs and a torso
$p := (p^1, p^2, p^3)^\top$	Generalized momentum
$x := (q^\top, p^\top)^\top$	State
$Q := (q^\top, \dot{q}^\top)^\top$	Angles and their velocities
$(q_{t^0}, p_{t^0}) := (q(t^0), p(t^0))$	Initial state
$(q_{t^1}, p_{t^1}) := (q(t^1), p(t^1))$	Terminal state
$(\cdot)_{-/+}$	Just before/after a discontinuous transition
	Note that $x_- \equiv x_{t^1}$.

in order to simplify the input–output relation in the Hamiltonian form mentioned later. Furthermore, we assume the following on this robot.

Assumption 1. *There exists a foot link whose thickness and mass can be ignored, and the ankle torque v^1 can be occurred relative to it.*

Assumption 2. *The foot of the swing leg does not bounce back nor slip on the ground at the collision (inelastic impulsive impact is assumed).*

Assumption 3. *Transfer of support between the stance and the swing legs is instantaneous.*

Assumption 4. *The foot-scuffing during the single support phase can be ignored.*

We use a number of notations with respect to the state. Table II summarizes them.

A typical mechanical system can be described by a Hamiltonian system in (1) with the state $x = (q^\top, p^\top)^\top \in \mathbb{R}^{2m}$ as

$$\begin{pmatrix} \dot{q} \\ \dot{p} \end{pmatrix} = \left(\begin{pmatrix} O_{mm} & I_m \\ -I_m & O_{mm} \end{pmatrix} - \begin{pmatrix} O_{mm} & O_{mm} \\ O_{mm} & R_D \end{pmatrix} \right) \begin{pmatrix} \frac{\partial H(q,p,u)^\top}{\partial q} \\ \frac{\partial H(q,p,u)^\top}{\partial p} \end{pmatrix},$$

$$y = -\frac{\partial H(q,p,u)^\top}{\partial u} = q, \qquad (21)$$

with the Hamiltonian

$$H(q, p, u) = \frac{1}{2} p^\top M(q)^{-1} p + U(q) - u^\top q. \quad (22)$$

Here, a positive definite matrix $M(q) \in \mathbb{R}^{m \times m}$ denotes the inertia matrix. The generalized momentum $p \in \mathbb{R}^m$ is given by $p := M(q)\dot{q}$. A positive semi-definite matrix $R_D \in \mathbb{R}^{m \times m}$ denotes the friction coefficients, and a scalar function $U(q) \in \mathbb{R}$ denotes the potential energy of the system. The dynamics of the robot depicted in Fig. 1 is described as a typical mechanical system in (21) with $m = 3$, the friction coefficients $R_D = O_{33}$ and the following inertia matrix and the potential energy:

$M(q)$
$$= \begin{pmatrix} m_T l^2 + m_L l^2 + m_L a^2 & -m_L bl \cos(q^1 - q^2) & m_T cl \cos(q^1 - q^3) \\ -m_L bl \cos(q^1 - q^2) & m_L b^2 & 0 \\ m_T cl \cos(q^1 - q^3) & 0 & m_T c^2 \end{pmatrix},$$

$U(q)$
$$= \{m_L((a+l)\cos q^1 - b \cos q^2) + m_T(l \cos q^1 + c \cos q^3)\}g.$$

The output y corresponding to the input u defined in Eq. (20) is given by $y = q$. At the end of a walking period, a collision between a leg and the ground causes a discontinuous change in angular velocities. Assumptions 2 and 3 imply that there exists no double support phase. Since the support and swing legs change each other instantly, we have

$$q_+ = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} q_- =: Cq_-, \quad (23)$$

where q_- and q_+ denote the angles just before and just after the collision, respectively (see Table 3). Following the law of conservation of the angular momentum, a transition mapping can be written as $\dot{q}_+ = \bar{\Pi}(q_-)\dot{q}_-$. The detail of the mapping $\bar{\Pi}(q_-)$ is omitted here. See e.g. ref. [12].

Before the ILC method mentioned above is applied, feedback controllers are typically employed to the control system in order to render the system asymptotically stable. However, the feedback system is not generally a Hamiltonian system of the form (1) any more with an arbitrary feedback controller. In ref. [32], a generalized canonical transformation, which is a pair of feedback and coordinate transformations preserving the Hamiltonian structure in (1), is proposed. It is known that in the case of a typical mechanical system in (21), a simple PD feedback preserves the structure of the Hamiltonian system.[22,32] Let us consider a PD controller

$$u = -K_P q - K_D \dot{q} + \bar{u}, \quad (24)$$

where $\bar{u}$ is a new input for ILC and $K_P, K_D \in \mathbb{R}^{m \times m}$ are symmetric positive definite matrices.

Remark 3 (ref. [22]). *Consider the feedback system of the form (21) by a PD controller (24). If the inertia matrix $M(q)$ of the system does not depend on the configuration coordinate q, then the conditions (3) and (4) in Lemma 1 are satisfied with the following nonsingular block diagonal matrix*

$T = \text{diag}\{I_m, -I_m\}$. *Otherwise, however, if PD gains K_P and K_D in (24) are chosen large enough, the conditions (3) and (4) are satisfied approximately with the same matrix T.*

In what follows, we consider the feedback system by a PD controller (24) with sufficiently large gains K_P and K_D so that the conditions in Lemma 1 are satisfied approximately, and derive the iteration law for the input $\bar{u}$ in Eq. (24).

4. Main results
This section proposes a repetitive control type optimal gait generation framework. In Section 4.1, we introduce a constraint by adding a virtual potential energy in order to prevent the robot from falling and to continue learning procedures. Then, the concept of the proposed framework is outlined. In Section 4.2, we propose a learning optimal control method of Hamiltonian systems by unifying ILC and IFT mentioned in Section 2.2. Since both the ILC and IFT methods influence each other, they regularly cannot be used simultaneously. The proposed method takes the interference of both methods into account. In Section 4.3, we define a cost function and exhibit a proposed algorithm.

4.1. Constraints by virtual potential energies
In the literature,[12,13] walking control methods using virtual constraints based on the output zeroing control are proposed. In ref. [13], particularly, they can achieve stable symmetric walking gaits by using another property of Hamiltonian systems other than those used in this paper. They set the output function $y = q^1 + q^2$ to zero by the output zeroing control and keep the leg angles bounded by a leg exchange scheme.[13] As a consequence, they guarantee that the robot does not fall and obtain symmetric walking gaits satisfying $q^1 + q^2 = 0$.

On the contrary, we use a similar idea of the virtual constraint to prevent the robot from falling, but do not use the output zeroing control. There are two reasons: one is that the output zeroing control requires the precise knowledge of the plant system and the other is that such constraints consume a large amount of control energy. We add a virtual potential energy P_c such as Eq. (25) to produce a similar effect to[13]

$$P_c := \frac{k_c}{2}(q^1 + q^2)^2. \quad (25)$$

Here, the gain parameter k_c represents the constraint strength. We let k_c sufficiently large at the beginning of the learning steps so that the trajectory of the robot is restricted to a symmetric one, i.e. $q^1 + q^2 = 0$ holds. Due to ref. [13], it is expected that the robot does not fall. The advantages of this method instead of the output zeroing control are as follows. First, it does not require the model parameters of the plant system, since the potential energy (25) can be generated by a simple feedback controller

$$u = -K_P q - K_D \dot{q} + \bar{u} - k_c A_c q, \quad A_c := \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \quad (26)$$
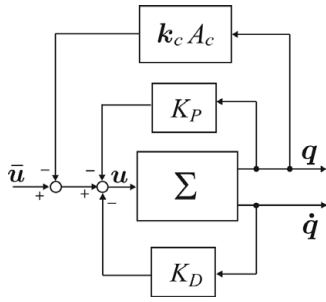
Fig. 2. Closed-loop system of the local PD feedback and the virtual potential one.

The feedback system is depicted in Fig. 2. Second, after adding the potential energy, the plant system preserves the Hamiltonian structure and the constraint parameter k_c is explicitly contained in a new Hamiltonian. The controller (26) converts the dynamics of the robot into another Hamiltonian system of the form (21) with a new Hamiltonian $\bar{H}$, a new structure matrix $\bar{J}$ and a new dissipation matrix $\bar{R}$ as

$$\bar{H}(q, p, \bar{u}, k_c)$$
$$= \frac{1}{2} p^\top M(q)^{-1} p + U(q) + \frac{1}{2} q^\top (K_P + k_c A_c) q - \bar{u}^\top q,$$
$$\bar{J} = \begin{pmatrix} O_{33} & I_3 \\ -I_3 & O_{33} \end{pmatrix}, \quad \bar{R} = \begin{pmatrix} O_{33} & O_{33} \\ O_{33} & K_D \end{pmatrix}. \tag{27}$$

By regarding k_c as a tuning parameter, we execute IFT mentioned in Section 2.2 to adjust the constraint strength, and we generate a walking trajectory by applying ILC simultaneously. The concept of the proposed framework is summarized as follows.

Step 1: Add a virtual potential energy to restrict the motion of the robot to a symmetric trajectory. Then, let the constraint parameter k_c sufficiently large to expect that the robot does not fall.

Step 2: By utilizing a unified learning optimal control method proposed in the next subsection, ILC generates an optimal walking gait and, simultaneously, IFT mitigates the constraint parameter automatically according to the progress of learning control.

Step 3: Repeat step 2 every one cycle of walking.

As a result, it is expected that an optimal gait without the constraint or with sufficiently small one is generated eventually. The feature of the proposed framework is that the robot keeps on walking and improves the walking gait, because the robot does not fall due to step 1. From this aspect, our method is classified as repetitive control[29] rather than ILC.[22, 25] It also differs from the conventional methods using virtual constraints in that it automatically optimizes the strength of the constraints.

4.2. Unified learning method of ILC and IFT

Here, we propose a learning optimal control method of Hamiltonian systems by unifying ILC[22] and IFT[26] by introducing an extended system which again has variational symmetry. This method enables one to execute ILC and IFT simultaneously, and plays an important role in our proposed framework mentioned in the previous subsection.

Let us define the extended input u_e by $u_e := (\bar{u}^\top, u_\rho^\top)^\top \in U_e = U \times U_\rho$, the extended output y_e by $y_e := (y^\top, y_\rho^\top)^\top \in Y_e = Y \times Y_\rho$, where $U_\rho, Y_\rho = L_2^s[t^0, t^1]$ and Hamiltonian H_e by $H_e(x, u_e) := H_c(x, \bar{u}, u_\rho)$. Then we have the following extended system $y_e = \Sigma_e^{x_{t^0}}(u_e)$:

$$\begin{cases} \dot{x} = (J - R) \dfrac{\partial H_e(x, u_e)^\top}{\partial x}, \quad x(t^0) = x_{t^0}, \\ y_e = -\dfrac{\partial H_e(x, u_e)^\top}{\partial u_e}. \end{cases} \tag{28}$$

Since the extended system (28) has the form of (1), it can be easily proved that this system has variational symmetry with certain conditions. Then, we consider a cost function $\hat{\Gamma}_e(u_e, y_e) : U_e \times Y_e \to \mathbb{R}$. The Fréchet derivative of the cost function can be calculated as

$$\delta\hat{\Gamma}_e(u_e, y_e)(\delta u_e, \delta y_e)$$
$$= \langle \nabla_{u_e} \hat{\Gamma}_e(u_e, y_e), \delta u_e \rangle_{U_e} + \langle \nabla_{y_e} \hat{\Gamma}_e(u_e, y_e), \delta y_e \rangle_{Y_e}$$
$$= \langle \nabla_{u_e} \hat{\Gamma}_e + (\delta\Sigma_e^{x_{t^0}}(u_e))^* (\nabla_{y_e} \hat{\Gamma}_e), \delta u_e \rangle_{U_e}, \tag{29}$$

where $\nabla_{u_e} \hat{\Gamma}_e(u_e, y_e)$ and $\nabla_{y_e} \hat{\Gamma}_e(u_e, y_e)$ represent the partial gradients of the cost function with respect to u_e and y_e, respectively. It follows from the definition of u_e and y_e and Eq. (17)

$$\delta u_e = \begin{pmatrix} \delta\bar{u} \\ \delta u_\rho \end{pmatrix} = \begin{pmatrix} \delta\bar{u} \\ \mathfrak{h}(\mathrm{d}\rho) \end{pmatrix}. \tag{30}$$

From Eq. (30), Eq. (29) reduces to

$$\delta\hat{\Gamma}_e(u_e, y_e)(\delta u_e, \delta y_e)$$
$$= \left\langle \begin{pmatrix} \mathrm{id} & 0 \\ 0 & \mathfrak{h}^* \end{pmatrix} \left(\nabla_{u_e} \hat{\Gamma}_e + (\delta\Sigma_e^{x_{t^0}}(u_e))^* (\nabla_{y_e} \hat{\Gamma}_e) \right), \begin{pmatrix} \delta\bar{u} \\ \mathrm{d}\rho \end{pmatrix} \right\rangle_{U \times \mathbb{R}^s}$$
$$= \left\langle \begin{pmatrix} \nabla_{\bar{u}} \hat{\Gamma}_e \\ \mathfrak{h}^*(\nabla_{u_\rho} \hat{\Gamma}_e) \end{pmatrix} + \begin{pmatrix} \mathrm{id} & 0 \\ 0 & \mathfrak{h}^* \end{pmatrix} \right.$$
$$\mathcal{R}\left(\delta\Sigma_e^{\psi_{t^0}}(w_e)\right) \mathcal{R}(\nabla_{y_e} \hat{\Gamma}_e), \left. \begin{pmatrix} \delta\bar{u} \\ \mathrm{d}\rho \end{pmatrix} \right\rangle_{U \times \mathbb{R}^s}$$
$$\approx \left\langle \begin{pmatrix} \nabla_{\bar{u}} \hat{\Gamma}_e \\ \mathfrak{h}^*(\nabla_{u_\rho} \hat{\Gamma}_e) \end{pmatrix} + \begin{pmatrix} \mathcal{R} & 0 \\ 0 & \mathfrak{h}^* \end{pmatrix} \right.$$
$$\times \left. \left(\frac{\Sigma_e^{\psi_{t^0}}(w_e + \epsilon_e \mathcal{R}(\nabla_{y_e} \hat{\Gamma}_e)) - \Sigma_e^{\psi_{t^0}}(w_e)}{\epsilon_e} \right), \begin{pmatrix} \delta\bar{u} \\ \mathrm{d}\rho \end{pmatrix} \right\rangle_{U \times \mathbb{R}^s}, \tag{31}$$

where ψ_{t^0} and $w_e := (w^\top, \mathfrak{h}(\rho)^\top)^\top$ should be chosen such that the condition (6) in Theorem 1 holds. In the last

approximation, the relation $\mathfrak{h}^*\mathcal{R} = \mathfrak{h}^*\mathcal{R}^* = (\mathcal{R}\mathfrak{h})^* = \mathfrak{h}^*$ is utilized (note that it follows from Eq. (7) that $\mathcal{R}^* = \mathcal{R}$). Consequently, the optimal learning control law unifying ILC and IFT is given by

$$
\begin{cases}
x_{t^0(3i+1)} = \psi_{t^0(i)}, \\
\bar{u}_{(3i+1)} = w_{(i)}, \\
\rho_{(3i+1)} = \rho_{(3i)},
\end{cases}
$$

$$
\begin{cases}
x_{t^0(3i+2)} = \psi_{t^0(i)}, \\
\bar{u}_{(3i+2)} = w_{(i)} + \epsilon_{e(i)}\mathcal{R}(\nabla_y \hat{\Gamma}_{e(3i)}), \\
u_{\rho(3i+2)} = \mathfrak{h}(\rho_{(3i)}) + \epsilon_{e(i)}\mathcal{R}(\nabla_{y_\rho} \hat{\Gamma}_{e(3i)}),
\end{cases}
\tag{32}
$$

$$
\begin{cases}
x_{t^0(3i+3)} = x_{t^0(3i)}, \\
\bar{u}_{(3i+3)} = \bar{u}_{(3i)} - K_{(i)}\left(\nabla_{\bar{u}} \hat{\Gamma}_{e(3i)} + \dfrac{1}{\epsilon_{e(i)}}\mathcal{R}(y_{(3i+2)} - y_{(3i+1)})\right), \\
\rho_{(3i+3)} = \rho_{(3i)} - K_{\rho(i)}\left(\displaystyle\int_{t^0}^{t^1} \nabla_{u_\rho} \hat{\Gamma}_{e(3i)} \right. \\
\qquad\qquad\qquad \left. + \dfrac{1}{\epsilon_{e(i)}}(y_{\rho(3i+2)} - y_{\rho(3i+1)})\mathrm{d}t\right),
\end{cases}
$$

provided that the initial control input $\bar{u}_{(0)} \equiv 0$ or an appropriate initial input, the initial parameter $\rho_{(0)}$ and the initial condition $x_{t^0(0)}$ are appropriately chosen, respectively. Here, $\epsilon_{e(\cdot)}$ denotes a sufficiently small positive constant and an appropriate positive definite matrices $K_{(\cdot)}$ and $K_{\rho(\cdot)}$ represent gains, respectively. Here, the condition $\psi_{e,t^0(i)}$ and $w_{(i)}$ are chosen such that it satisfies the condition (6) in Theorem 1 with the trajectory governed by the pair of $x_{t^0(3i)}$ and $\bar{u}_{(3i)}$ with $\rho_{(3i)}$. A concrete algorithm exhibiting how to select $\psi_{e,t^0(i)}$ and $w_{(i)}$ is given for mechanical systems in refs. [21, 26].

We supplement the derivation of the control law (32). The triple iteration laws imply that this learning procedure needs three experiments to execute a single update in (12). First, the $(3i+1)$th iteration generates a trajectory ψ corresponding to $\phi = x_{(3i)}$ such that the condition (6) in Theorem 1 holds. Second, in the $(3i+2)$th iteration, we calculate the output $\Sigma_e^{\psi_{t^0}}(w_e + \epsilon_e\mathcal{R}(\nabla_{y_e}\hat{\Gamma}_e))$ in Eq. (31) (note that in this case ψ corresponds to $x_{(3i+1)}$). It follows from $w_e = (w^\top, \mathfrak{h}(\rho)^\top)^\top$ and $\nabla_{y_e}\hat{\Gamma}_e = (\nabla_y\hat{\Gamma}_e^\top, \nabla_{y_\rho}\hat{\Gamma}_e^\top)^\top$ that

$$
w_e + \epsilon_e\mathcal{R}(\nabla_{y_e}\hat{\Gamma}_e) = \begin{pmatrix} w + \epsilon_e\mathcal{R}(\nabla_y\hat{\Gamma}_e) \\ \mathfrak{h}(\rho) + \epsilon_e\mathcal{R}(\nabla_{y_\rho}\hat{\Gamma}_e) \end{pmatrix}.
\tag{33}
$$

Note that $\bar{u}_{(3i+2)}$ and $u_{\rho(3i+2)}$ follows from Eq. (33). Then the input and the output signals of $\delta\Sigma_e^{x_{t^0}}(u_e)^*(\nabla_{y_e}\hat{\Gamma}_e)$ can be calculated from the last approximation in Eq. (31). With this information, the gradient of the cost function with respect to the input $\nabla\Gamma_e(u_e)$ with $\Gamma_e(u_e) := \hat{\Gamma}_e(u_e, \Sigma_e(u_e))$(see also Eq. (11)) is obtained. Finally, the input for the $(3i+3)$th iteration is given by Eq. (12) with these signals. That is, the $(3i+3)$th iteration law comes from the following

calculation:

$$
\begin{aligned}
\begin{pmatrix} \delta\bar{u}_{(3i+3)} \\ \mathrm{d}\rho_{(3i+3)} \end{pmatrix} &:= \begin{pmatrix} \bar{u}_{(3i+3)} - \bar{u}_{(3i)} \\ \rho_{(3i+3)} - \rho_{(3i)} \end{pmatrix} \\
&= -\begin{pmatrix} K_{(i)} & 0 \\ 0 & K_{\rho(i)} \end{pmatrix}\nabla\Gamma_e(u_{e(3i)}) \\
&\approx -\begin{pmatrix} K_{(i)} & 0 \\ 0 & K_{\rho(i)} \end{pmatrix}\left\{\begin{pmatrix} \nabla_{\bar{u}}\hat{\Gamma}_{e(3i)} \\ \mathfrak{h}^*(\nabla_{u_\rho}\hat{\Gamma}_{e(3i)}) \end{pmatrix}\right. \\
&\qquad \left. + \dfrac{1}{\epsilon_e}\begin{pmatrix} \mathcal{R} & 0 \\ 0 & \mathfrak{h}^* \end{pmatrix}\begin{pmatrix} y_{(3i+2)} - y_{(3i+1)} \\ y_{\rho(3i+2)} - y_{\rho(3i+1)} \end{pmatrix}\right\}.
\end{aligned}
$$

Remark 4. *If the learning procedure is executed around a symmetric trajectory and a trajectory in each experiment approximately satisfies the condition (9), then one can utilize the following procedure instead of that in (32):*

$$
\begin{cases}
x_{t^0(2i+1)} = x_{t^0(2i)}, \\
\bar{u}_{(2i+1)} = \bar{u}_{(2i)} + \epsilon_{e(i)}\mathcal{R}(\nabla_y\hat{\Gamma}_{e(2i)}), \\
u_{\rho(2i+1)} = \mathfrak{h}(\rho_{(2i)}) + \epsilon_{e(i)}\mathcal{R}(\nabla_{y_\rho}\hat{\Gamma}_{e(2i)}),
\end{cases}
$$

$$
\tag{34}
$$

$$
\begin{cases}
x_{t^0(2i+2)} = x_{t^0(2i)}, \\
\bar{u}_{(2i+2)} = \bar{u}_{(2i)} - K_{(i)}\left(\nabla_{\bar{u}}\hat{\Gamma}_{e(2i)} + \dfrac{1}{\epsilon_{e(i)}}\mathcal{R}(y_{(2i+1)} - y_{(2i)})\right), \\
\rho_{(2i+2)} = \rho_{(2i)} - K_{\rho(i)}\left(\displaystyle\int_{t^0}^{t^1} \nabla_{u_\rho}\hat{\Gamma}_{e(2i)} \right. \\
\qquad\qquad\qquad \left. + \dfrac{1}{\epsilon_{e(i)}}(y_{\rho(2i+1)} - y_{\rho(2i)})\mathrm{d}t\right).
\end{cases}
$$

Around a symmetric trajectory, the condition (9) is satisfied with $\psi = \phi$ and $w = v$ due to Remark 2. Here, ϕ represents the state x and v represents the learning input $\bar{u}$. So, the procedure corresponding to the $(3i+1)$th iteration in (32) is not necessary. The learning law (34) follows from that the $(2i+1)$th and the $(2i+2)$th procedures correspond to the $(3i+2)$th and the $(3i+3)$th procedures in (32), respectively.

Remark 5. *Although the proposed algorithm requires time-varying feedback gains during the learning procedure in generating perturbation signals (see $u_{\rho(3i+2)}$ in the iteration procedure (32) or $u_{\rho(2i+1)}$ in another procedure (34) in order to approximate output signals of the variational system by utilizing Eq. (31), eventually, a generated optimal feedback gain is constant. Unless time-varying feedback gains are available, we substitute a feedforward input with the previous output signal. For example, the following procedure is substituted for the original one in (32):*

$$
\begin{cases}
x_{t^0(3i+2)} = \psi_{t^0(i)}, \\
\bar{u}_{(3i+2)} = w_{(i)} + \epsilon_{e(i)}\left(\mathcal{R}(\nabla_y\hat{\Gamma}_{e(3i)}) + \mathcal{R}(\nabla_{y_\rho}\hat{\Gamma}_{e(3i)})y_{(3i)}\right), \\
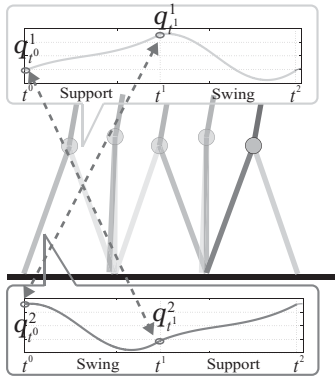\rho_{(3i+2)} = \rho_{(3i)}.
\end{cases}
$$

Fig. 3. Illustration of the restraint condition of the cost function.

4.3. Optimal gait generation algorithm

Let us consider the following cost function:

$$
\begin{aligned}
&\hat{\Gamma}(y, \dot{y}, \bar{u}, y_\rho, u_\rho) \\
&:= \frac{1}{2} \int_{t^0}^{t^1} (y(\tau) - C\mathcal{R}(y)(\tau))^\top \nu_1(\tau) \Lambda_y (y(\tau) - C\mathcal{R}(y)(\tau)) d\tau \\
&\quad + \frac{1}{2} \int_{t^0}^{t^1} F_v(\dot{y}(\tau) - v_{\text{ref}})^\top \nu_2(\tau) \Lambda_{\dot{y}} F_v(\dot{y}(\tau) - v_{\text{ref}}) d\tau \\
&\quad + \frac{1}{2} \int_{t^0}^{t^1} \bar{u}(\tau)^\top \Lambda_{\bar{u}} \bar{u}(\tau) d\tau + \frac{\gamma_{y_\rho}}{2} \int_{t^0}^{t^1} y_\rho^2(\tau) d\tau \\
&\quad + \frac{\gamma_{u_\rho}}{2} \int_{t^0}^{t^1} u_\rho^2(\tau) d\tau,
\end{aligned}
\tag{35}
$$

where appropriate positive definite matrices $\Lambda_y, \Lambda_{\dot{y}}, \Lambda_{\bar{u}} \in \mathbb{R}^{3 \times 3}$ represent weight matrices and appropriate positive constants γ_{y_ρ} and γ_{u_ρ} represent weighting coefficients, respectively. The first term of the cost function (35) is also equipped in refs. [19, 30], which is a necessary condition for a periodic trajectory such that $q^1(t^0) \equiv q^2(t^1)$ and $q^2(t^0) \equiv q^1(t^1)$. Figure 3 illustrates the condition. Let us note that although another necessary condition with respect to $\dot{q}$ can be utilized as in ref. [19], where initial angular velocities are equivalent to velocities just after touchdown. However, it is not equipped here for simplicity of the iteration law. In the second term, $v_{\text{ref}} \in \mathbb{R}^3$ represents a constant reference angular velocity, $\nu_1(t) \in \mathbb{R}$ and $\nu_2(t) \in \mathbb{R}$ denote filter functions defined respectively by

$$
\nu_1(t) := \begin{cases} \frac{1}{2}\left(1 - \cos\left(\frac{t^0 + \Delta t - t}{\Delta t}\pi\right)\right) & (t^0 \le t \le t^0 + \Delta t) \\ 0 & (t^0 + \Delta t < t \le t^1), \end{cases}
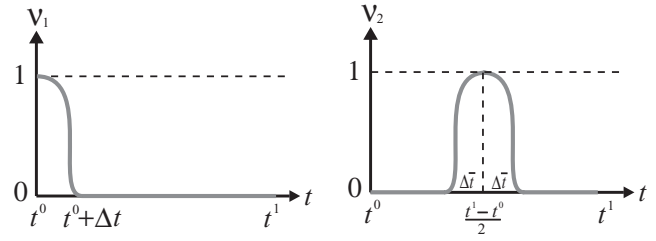\tag{36}
$$

Fig. 4. Filter functions ν_1 and ν_2.

$$
\nu_2(t) :=
$$

$$
\times \begin{cases} 0 & \left(t^0 \le t < \frac{t^1 - t^0}{2} - \Delta\bar{t}\right), \\ \frac{1}{2}\left(1 - \cos\left(\frac{-\frac{t^1-t^0}{2} + \Delta\bar{t} + t}{\Delta\bar{t}}\pi\right)\right) & \left(\frac{t^1-t^0}{2} - \Delta\bar{t} \le t < \frac{t^1-t^0}{2}\right), \\ \frac{1}{2}\left(1 - \cos\left(\frac{\frac{t^1-t^0}{2} + \Delta\bar{t} - t}{\Delta\bar{t}}\pi\right)\right) & \left(\frac{t^1-t^0}{2} \le t < \frac{t^1-t^0}{2} + \Delta\bar{t}\right), \\ 0 & \left(\frac{t^1-t^0}{2} + \Delta\bar{t} \le t \le t^1\right), \end{cases}
\tag{37}
$$

where design parameters Δt and $\Delta\bar{t}$ denote positive constants. Figure 4 illustrates $\nu_1(t)$ and $\nu_2(t)$. For any $\zeta \in \mathbb{R}^r$, a penalty function $F_v : \mathbb{R}^r \to \mathbb{R}^r$ is defined as

$$
[F_v(\zeta)]^i = \begin{cases} k_{F_v}(\zeta^i)^2 & \text{if } \zeta^i < 0, \\ 0 & \text{otherwise,} \end{cases} \quad (i = 1, 2, \ldots, r),
\tag{38}
$$

where an appropriate positive constant k_{F_v} represents strength of the penalty. In what follows, the dimension r of the penalty function shall accordingly change with that of the argument (in the case of Eq. (35), $r = 3$). The second term encourages the robot to achieve an appropriate constant velocity in the middle of walking. As a consequence, it is aimed at specifying the walking direction (forward or backward) and a rough walking speed, and preventing the robot from stopping during the learning. The third and fourth terms are to minimize the control input and the feedback input generating virtual potential energy, respectively. The last term is to mitigate the strength of the virtual constraint.

In order to utilize variational symmetry of the extended system for (27), let us rewrite the cost function (35) as

$$
\begin{aligned}
&\hat{\Gamma}(y, \dot{y}, \bar{u}, y_\rho, u_\rho) \\
&\equiv \frac{1}{2} \int_{t^0}^{t^1} (y_e(\tau) - C_e \mathcal{R}(y_e)(\tau))^\top \Lambda_{y_e}(\tau) \\
&\quad \times (y_e(\tau) - C_e \mathcal{R}(y_e)(\tau)) d\tau \\
&\quad + \frac{1}{2} \int_{t^0}^{t^1} F_v(\dot{y}_e(\tau) - v_{e,\text{ref}})^\top \Lambda_{\dot{y}_e}(\tau) F_v(\dot{y}_e(\tau) - v_{e,\text{ref}}) d\tau \\
&\quad + \frac{1}{2} \int_{t^0}^{t^1} u_e(\tau)^\top \Lambda_{u_e} u_e(\tau) d\tau =: \hat{\Gamma}_e(y_e, \dot{y}_e, u_e),
\end{aligned}
\tag{39}
$$

where

$$C_e := \text{diag}\{C, 0\} \in \mathbb{R}^{4 \times 4}, \quad v_{e,\text{ref}} := (v_{\text{ref}}, 0)^\top \in \mathbb{R}^4,$$

$$\Lambda_{y_e}(t) := \text{diag}\{\nu_1(t)\Lambda_y, \gamma_{y_\rho}\}, \quad \Lambda_{\dot{y}_e}(t) := \text{diag}\{\nu_2(t)\Lambda_{\dot{y}}, 0\},$$

$$\Lambda_{u_e} := \text{diag}\{\Lambda_{\bar{u}}, \gamma_{u_\rho}\} \in \mathbb{R}^{3 \times 3} \tag{40}$$

and, in this case, the dimension of F_v is $r = 4$.

Since the virtual constraint introduced in Section 4.1 restricts the motion of the robot to a symmetric trajectory, it is supposed that the learning procedure is executed around a symmetric trajectory. Now let us derive the concrete updating law based on the optimal learning control procedure (34) in Remark 4.

Let us calculate the Fréchet derivative of the cost function (39) as follows:

$$\delta \hat{\Gamma}_e(y_e, \dot{y}_e, u_e)(\delta y_e, \delta \dot{y}_e, \delta u_e)$$
$$= \langle \Lambda_{y_e}(y_e - C_e \mathcal{R}(y_e)), \delta y_e - C_e \mathcal{R}(\delta y_e) \rangle_{Y_e}$$
$$\quad + \langle \Lambda_{\dot{y}_e} F_v(\dot{y}_e - v_{e,\text{ref}}), \delta F_v(\dot{y}_e - v_{e,\text{ref}}) \mathcal{D}_t(\delta y_e) \rangle_{Y_e}$$
$$\quad + \langle \Lambda_{u_e} u_e, \delta u_e \rangle_{U_e}$$
$$= \langle (\text{id} - \mathcal{R}C_e)\Lambda_{y_e}(\text{id} - C_e \mathcal{R})(y_e)$$
$$\quad - \mathcal{D}_t \left((\delta F_v(\dot{y}_e - v_{e,\text{ref}}))^* \Lambda_{\dot{y}_e} \right.$$
$$\quad \left. \times F_v(\dot{y}_e - v_{e,\text{ref}})\right), \delta y_e \rangle_{Y_e} + \langle \Lambda_{u_e} u_e, \delta u_e \rangle_{U_e}$$
$$=: \langle \nabla_{y_e} \hat{\Gamma}_e, \delta y_e \rangle_{Y_e} + \langle \nabla_{u_e} \hat{\Gamma}_e, \delta u_e \rangle_{U_e}$$
$$= \left\langle \begin{pmatrix} \nabla_y \hat{\Gamma}_e \\ \nabla_{y_\rho} \hat{\Gamma}_e \end{pmatrix}, \begin{pmatrix} \delta y \\ \delta y_\rho \end{pmatrix} \right\rangle_{Y_e} + \left\langle \begin{pmatrix} \nabla_{\bar{u}} \hat{\Gamma}_e \\ \nabla_{u_\rho} \hat{\Gamma}_e \end{pmatrix}, \begin{pmatrix} \delta \bar{u} \\ \delta u_\rho \end{pmatrix} \right\rangle_{U_e}. \tag{41}$$

Here, $\mathcal{D}_t$ denotes the time derivative operator. Since the cost function (39) is a functional of the time derivative of the output $\dot{y}_e$, the previously proposed technique with a pseudo-adjoint of $\mathcal{D}_t$ in ref. [19] is utilized in the calculation. Briefly put, $\mathcal{D}_t^* = -\mathcal{D}_t$ holds under certain conditions. See refs. [19, 21] for the details. From Eq. (38), $(\delta F_v(\dot{y}_e - v_{e,\text{ref}}))^*$ can be calculated as

$$[(\delta F_v(\dot{y}_e - v_{e,\text{ref}}))^*]_j^i$$
$$= \begin{cases} 0 & (i \neq j) \\ \begin{cases} 2k_{F_v} \left(\dot{y}_e^i - v_{e,\text{ref}}^i\right) & \text{if } \dot{y}_e^i - v_{e,\text{ref}}^i < 0 \\ 0 & \text{otherwise} \end{cases} & (i = j) \end{cases},$$
$$(i, j = 1, 2, 3, 4). \tag{42}$$

Then, the partial gradients $\nabla_y \hat{\Gamma}_e$, $\nabla_{y_\rho} \hat{\Gamma}_e$, $\nabla_{\bar{u}} \hat{\Gamma}_e$ and $\nabla_{u_\rho} \hat{\Gamma}_e$ are calculated from Eq. (41) as

$$\begin{pmatrix} \nabla_y \hat{\Gamma}_e \\ \nabla_{y_\rho} \hat{\Gamma}_e \end{pmatrix}$$
$$= \begin{pmatrix} \text{id} - \mathcal{R}C & 0 \\ 0 & \text{id} \end{pmatrix} \begin{pmatrix} \nu_1 \Lambda_y & O_{21} \\ O_{12} & \gamma_{y_\rho} \end{pmatrix} \begin{pmatrix} \text{id} - C\mathcal{R} & 0 \\ 0 & \text{id} \end{pmatrix}$$
$$\times \begin{pmatrix} y \\ y_\rho \end{pmatrix} - \mathcal{D}_t \left(\begin{pmatrix} (\delta F_v(\dot{y} - v_{\text{ref}}))^* & 0 \\ 0 & (\delta F_v(\dot{y}_\rho - 0))^* \end{pmatrix} \right.$$
$$\times \left. \begin{pmatrix} \nu_2 \Lambda_{\dot{y}} & O_{21} \\ O_{12} & 0 \end{pmatrix} \begin{pmatrix} F_v(\dot{y} - v_{\text{ref}}) \\ F_v(\dot{y}_\rho - 0) \end{pmatrix} \right)$$
$$= ((\text{id} - \mathcal{R}C)\nu_1 \Lambda_y(\text{id} - C\mathcal{R})(y) - \mathcal{D}_t((\delta F_v(\dot{y} - v_{\text{ref}}))^* \nu_2$$
$$\times \Lambda_{\dot{y}} F_v(\dot{y} - v_{\text{ref}})), \ \gamma_{y_\rho} y_\rho)^\top \tag{43}$$

and

$$\begin{pmatrix} \nabla_{\bar{u}} \hat{\Gamma}_e \\ \nabla_{u_\rho} \hat{\Gamma}_e \end{pmatrix} = \begin{pmatrix} \Lambda_{\bar{u}} \bar{u} \\ \gamma_{u_\rho} u_\rho \end{pmatrix}. \tag{44}$$

From the iteration law (34) and Eqs. (43) and (44), let us summarize the proposed learning algorithm.

Step 0: Set appropriate positive definite matrices Λ_y, $\Lambda_{\dot{y}}$ and $\Lambda_{\bar{u}}$ as weight matrices, positive constants γ_{y_ρ} and γ_{u_ρ} as weight coefficients and positive constants Δt, $\Delta \bar{t}$ and k_{F_v} as design parameters for the filter functions ν_1 in (36) and ν_2 in (37) and the penalty function F_v in (38). Set the initial control input $\bar{u}_{(0)}$ appropriately (or set $\bar{u}_{(0)} \equiv 0$) and a constant reference angular velocity v_{ref} and let the constraint parameter $k_{c(0)}$ sufficiently large. Let the robot start walking under an appropriate initial condition x_{t^0}. Set $i = 0$. Then, go to step 1.

Step $2i + 1$: During the $(2i+1)$th walking cycle, one utilizes the following controller:

$$u = -K_P q - K_D \dot{q} - u_{\rho(2i+1)} A_c q + \bar{u}_{(2i+1)}. \tag{45}$$

Here, the time-varying feedback gain for the virtual constraint $u_{\rho(2i+1)}$ and the feedforward control input $\bar{u}_{(2i+1)}$ are given by

$$\begin{cases} \bar{u}_{(2i+1)} = \bar{u}_{(2i)} + \epsilon_{e(i)} \mathcal{R}(\nabla_y \hat{\Gamma}_{e(2i)}), \\ u_{\rho(2i+1)} = u_{\rho(2i)} + \epsilon_{e(i)} \mathcal{R}(\nabla_{y_\rho} \hat{\Gamma}_{e(2i)}), \end{cases} \tag{46}$$

where $\epsilon_{e(i)}$ denotes a sufficiently small positive constant and

$$\nabla_y \hat{\Gamma}_{e(2i)} = (\text{id} - \mathcal{R}C)\nu_1 \Lambda_y(\text{id} - C\mathcal{R})(y_{(2i)})$$
$$\quad - \mathcal{D}_t((\delta F_v(\dot{y}_{(2i)} - v_{\text{ref}}))^* \nu_2 \Lambda_{\dot{y}} F_v(\dot{y}_{(2i)} - v_{\text{ref}})),$$

$$\nabla_{y_\rho} \hat{\Gamma}_{e(2i)} = \gamma_{y_\rho} y_{\rho(2i)}.$$

For $(\delta F_v)^*$, see Eq. (42). Then go to step $2i+2$.

Step 2i + 2: During the $(2i+2)$th walking cycle, one utilizes the following controller:

$$u = -K_P q - K_D \dot{q} - k_{c(2i+2)} A_c q + \bar{u}_{(2i+2)}. \quad (47)$$

Here the feedback gain $k_{c(2i+2)}$ which represents the strength of the virtual constraint and the feedforward control input $\bar{u}_{(2i+2)}$ are given by

$$\begin{cases} \bar{u}_{(2i+2)} = \bar{u}_{(2i)} - K_{(i)} \\ \quad \times \left(\Lambda_{\bar{u}} \bar{u}_{(2i)} + \dfrac{1}{\epsilon_{e(i)}} \mathcal{R}(y_{(2i+1)} - y_{(2i)}) \right), \\ k_{c(2i+2)} = k_{c(2i)} - K_{\rho(i)} \left(\gamma_{u_\rho} k_{c(2i)}(t^1 - t^0) \right. \\ \quad \left. + \dfrac{1}{\epsilon_{e(i)}} \displaystyle\int_{t^0}^{t^1} y_{\rho(2i+1)} - y_{\rho(2i)} \mathrm{dt} \right), \end{cases} \quad (48)$$

where appropriate positive definite matrix $K_{(i)}$ and positive constant $K_{\rho(i)}$ represent learning and tuning gains, respectively. Set $i = i + 1$ and go to step $2i + 1$.

Here are all assumptions in the proposed framework and how those are satisfied.

1. *Conditions* (3) *and* (4) *in Lemma 1*
   If the inertia matrix $M(q)$ of the robot does not depend on the configuration coordinate q, then both conditions are satisfied with the nonsingular block diagonal matrix $T = \mathrm{diag}\{I_m, -I_m\}$. Otherwise, however, if PD gains K_P and K_D in the PD controller (26) are chosen large enough, both conditions are satisfied approximately with the same matrix T. For the proof, see ref [22].
2. *Difference approximation of the variational system in* (13) *(see also* (31)*)*
   In order to let the approximation hold, we can make the input variation arbitrarily small by choosing a sufficiently small constant $\epsilon_e > 0$.
3. *Restriction to a symmetric walking trajectory*
   In the beginning of learning, we make the constraint parameter $k_{c(0)}$ sufficiently large.

5. Numerical examples

We apply the proposed algorithm in the previous section to the compass gait biped with a torso depicted in Fig. 1 in order to generate an optimal periodic gait. Here, we show the results of two kinds of simulations. The physical parameters of the robot in Table I are chosen as $m_T = 5.0$, $m_L = 1.2$ (kg) and $a = b = 0.20$, $c = 0.12$ (m), which are the same as those of the robot named Skipper II in ref. [13]. For the controller, the following feedback gains are utilized $K_P = \mathrm{diag}(4, 4, 6)$ and $K_D = \mathrm{diag}(2, 2, 4)$. In all simulations, we assign a reference velocity only to $\dot{q}^1$, since the angular velocity of the ankle joint of the support leg $\dot{q}^1$ relates to leaning forward of the body and mainly affects the walking velocity. This is the reason why we do not assign the reference velocity to the center of mass (CoM) of the robot, that its calculation requires the precise knowledge of the robot model, e.g. the inertia matrix. We utilize the following design parameters with respect to weighting functions for the cost
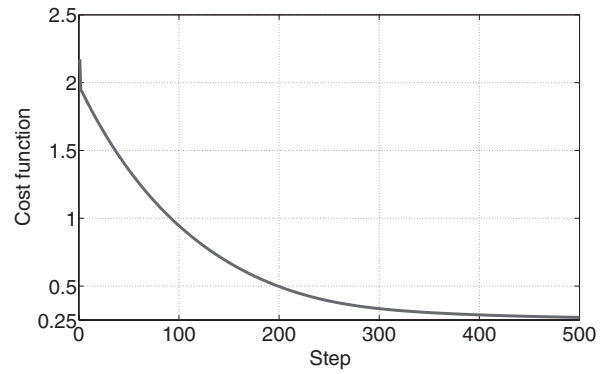

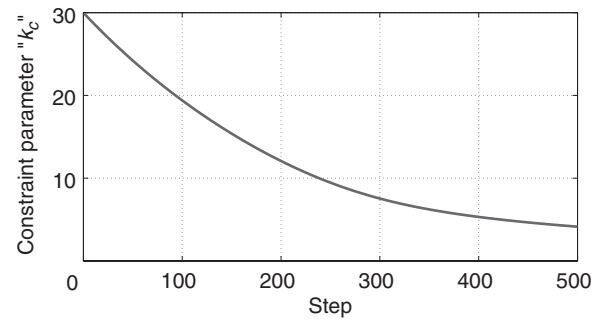Fig. 5. Histories of the cost function $\hat{\Gamma}_e$ along walking steps.

Fig. 6. Histories of the constraint parameter k_c along walking steps.

function (39) as $\Lambda_y = \mathrm{diag}(20, 20, 20)$, $\Lambda_{\dot{y}} = \mathrm{diag}(10, 0, 0)$, $\gamma_{y_\rho} = 1 \times 10^{-2}$, $\Lambda_{\bar{u}} = \mathrm{diag}(1 \times 10^{-4}, 5 \times 10^{-5}, 5 \times 10^{-5})$ and $\gamma_{u_\rho} = 1 \times 10^{-2}$, those with respect to filter functions and penalty function as $\Delta t = 5.0 \times 10^{-3}$(s), $\Delta \bar{t} = 0.1$ (s) and $k_{F_v} = 0.25$ and those with respect to the learning algorithm as $K_{(\cdot)} = \mathrm{diag}(3, 3, 3)$, $K_{\rho(\cdot)} = 1$ and $\epsilon_{e(\cdot)} = 1$. In each simulation, we proceed 500 steps of the learning procedure, which means the robot continued 1000 cycles of walking, with the initial constraint parameter $k_{c(0)} = 30$, with the initial condition:

$$\left(q_{t^0}^1, q_{t^0}^2, q_{t^0}^3, \dot{q}_{t^0}^1, \dot{q}_{t^0}^2, \dot{q}_{t^0}^3\right) = (-0.18, 0.20, 0, 1.1, 0.5, 0),$$

and with the initial control input:

$$\left(\bar{u}_{(0)}^1(t), \bar{u}_{(0)}^2(t), \bar{u}_{(0)}^3(t)\right)^\top \equiv (0.5, -1.5, 0)^\top. \quad (49)$$

In the first simulation, we assign a reference velocity in (40) as $v_{\mathrm{ref}} = (0.5, 0, 0)^\top$. Figure 5 shows the history of the cost function (39) along the walking steps. Since the cost function monotonically decreases along the walking steps and then converges to a constant value, it implies that at least a local minimum of the cost function has been achieved smoothly. Figure 6 shows the history of the constraint parameter k_c along the walking steps. It implies that the strength of the virtual constraint is adjusted. Although k_c does not converge to zero, it plays a role of a stabilizing feedback controller. Figures 7 and 8 represent the animations of the robot in the first and the last five cycles of walking, respectively. These figures show that at the beginning the robot walks awkwardly, and then the robot improves the walking gait as it continues to walk.
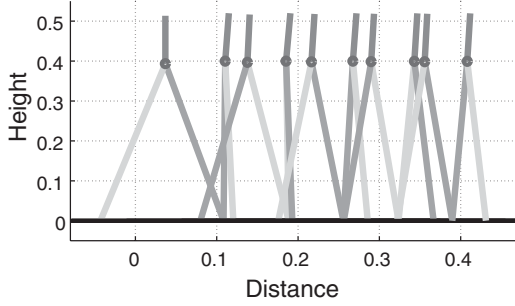
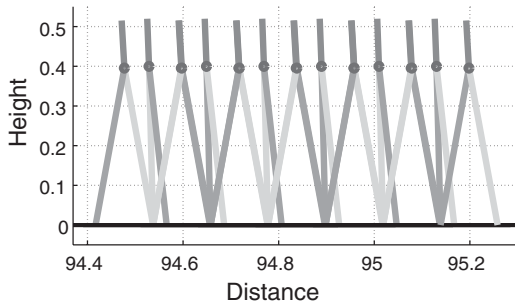Fig. 7. Stick diagrams in the first five cycles of walking.

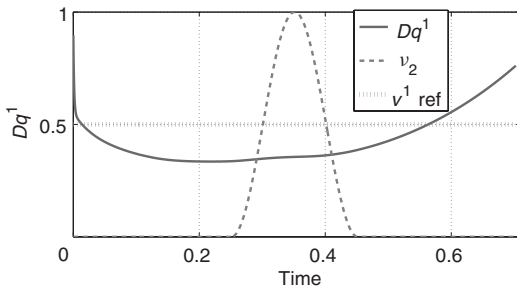Fig. 8. Stick diagrams in the last five cycles of walking.

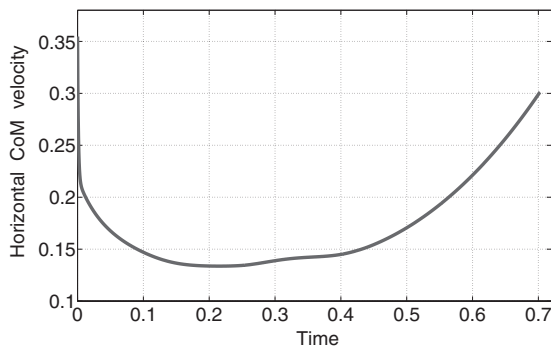Fig. 9. $\dot{q}^1$, v_{ref}^1 and v_2 of the generated gait.

Fig. 10. Horizontal velocity of the center of mass (CoM) of the generated gait.

Figure 9 shows $\dot{q}^1$ of the generated gait, its reference $v_{\mathrm{ref}}^1 = 0.5$ and the filter function $v_2(t)$ in (37), and Fig. 10 shows the horizontal velocity of CoM of the generated gait. In the next numerical example, we will observe that the horizontal CoM velocity increases with the larger reference velocity. Figure 11 exhibits the phase portraits of $q - \dot{q}$. The fact that a periodic trajectory is generated follows from that the phase portraits in the figure form closed orbits.
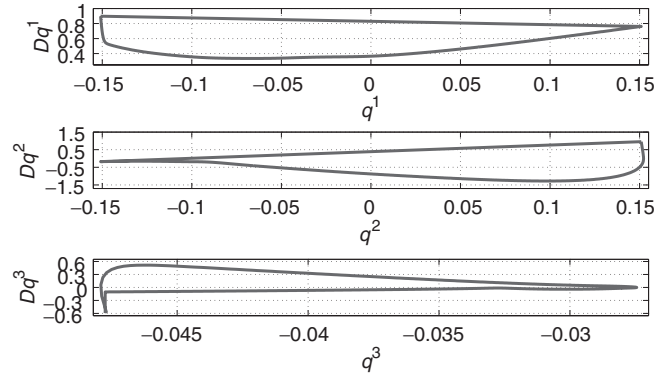
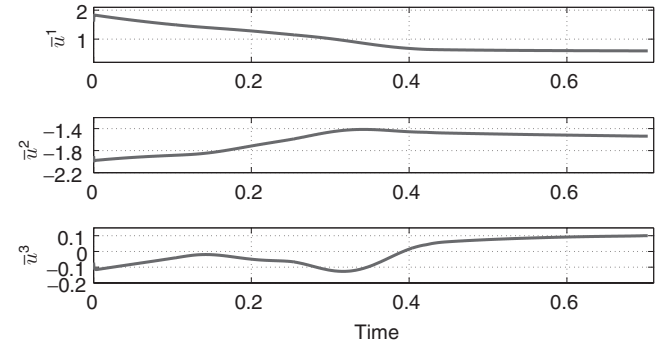Fig. 11. Projections of the phase portrait to $q_i - \dot{q}_i$ planes ($i = 1, 2, 3$).

Fig. 12. Generated learning control inputs $\bar{u}$.

Figure 12 shows the generated learning control input $\bar{u}$, and Fig. 13 shows the control input u in Eq. (47). Finally, we quantitatively evaluate the energy efficiency of the generated gait. Here, we calculate the specific resistance (SR),[33] which is defined as $\mathrm{SR} = \bar{E}/(M_{\mathrm{all}} g \bar{V})$, where $\bar{E}$ represents the average input power, M_{all} represents the total mass and $\bar{V}$ represents the average walking speed. In the case of the compass robot with a torso, $\bar{E}$ is obtained by[34] (for the control input u, see Eq. (26))

$$\bar{E} = \frac{1}{t^1 - t^0} \int_{t^0}^{t^1} |u^1(t) q^1(t)| + |u^2(t) q^2(t)| + |u^3(t) q^3(t)| \mathrm{d}t.$$

From the above definition, we have $\mathrm{SR} = 0.0480$. It implies that we can achieve almost equivalent efficiency to a human walking according to ref. [33]. Besides, we also investigate the dimensionless index called the cost of transport (CT),[35] which is calculated by $\mathrm{CT} = E/(M_{\mathrm{all}} \Delta X_g)$. Here, the used energy E is calculated by $E = \int_{t^0}^{t^1} |u(t)^\top q(t)| \mathrm{d}t$, and ΔX_g represents the travel distance. We have $\mathrm{CT} = 0.435$.

In the second simulation, we assign another reference velocity in (40) as $v_{\mathrm{ref}} = (0.8, 0, 0)^\top$. The purpose of the second example is to show that we can change the speed of a generated gait by only changing the reference velocity v_{ref}. Although we have observed that a various walking trajectories can be generated by changing other parameters, e.g. weight matrices of the cost function, we fix the other parameters except for v_{ref} in order to make it clear whether the reference velocity changes the speed of the generated gait or not. Due to the limitation of space, we prioritize the
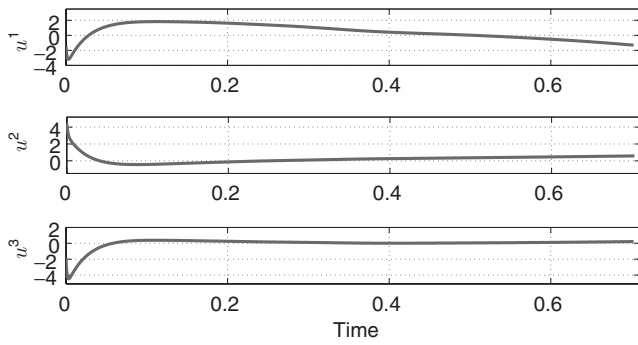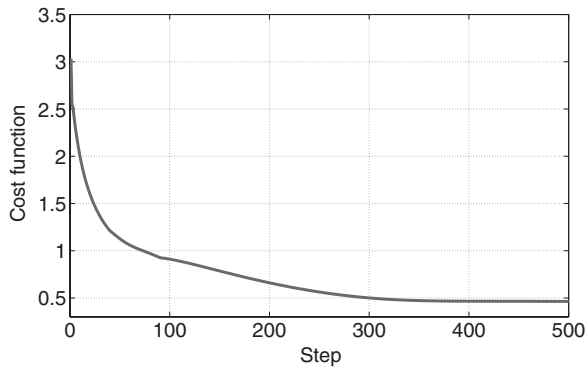
Fig. 13. Generated all control inputs u.
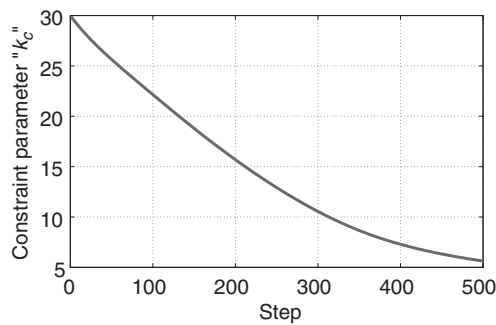

Fig. 14. Cost function.

Fig. 15. Constraint parameter k_c.

Fig. 16. Stick diagrams in the first five cycles of walking.

Fig. 17. Stick diagrams in the last five cycles of walking.

Fig. 18. Phase portrait of $q - \dot{q}$.
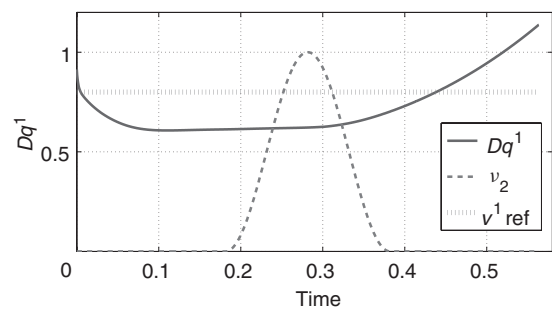
Fig. 19. $\dot{q}^1$, v_{ref}^1 and v_2 of the generated gait.

comparison of gait speed with different reference velocities over the other simulation results. Figure 14 shows the history of the cost function (39) along the walking steps, and it implies that at least a local minimum of the cost function has been achieved smoothly as in the first simulation results. Figure 15 shows the history of the constraint parameter k_c along the walking steps, and the strength of the virtual
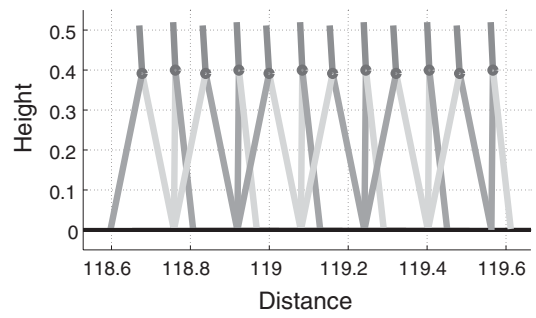
constraint is adjusted. Figures 16 and 17 represent the animations of the robot in the first and the last five cycles of walking, respectively. These figures show that an optimal gait can be eventually generated as well as the first simulation. We also confirm that the phase portraits corresponding to the generated gait depicted in Fig. 18 form closed orbits, which implies that the resultant gait is periodic one. Figure 19 shows $\dot{q}^1$ of the generated gait, its reference $v_{\text{ref}}^1 = 0.8$ and the filter function $v_2(t)$, and Fig. 20 shows the horizontal velocity of CoM of the generated gait. Since the assigned reference velocity here is bigger than that in the first simulation, the velocity of CoM of the generated gait is faster than that in the first one. From these simulation results, the horizontal velocity of CoM has a relation to $\dot{q}^1$, and the second term of the cost function (39) with appropriately chosen $v_{e,\text{ref}}$ encourages the robot to walk forward with appropriate velocity.
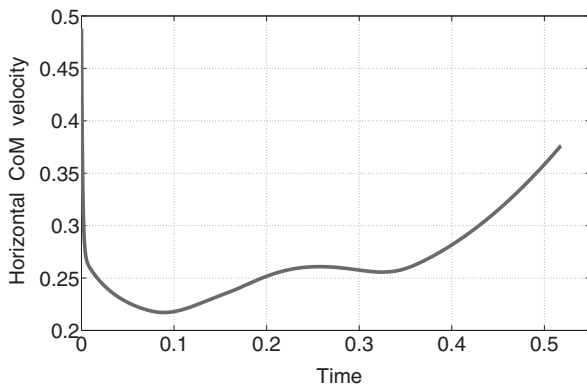
Fig. 20. Horizontal velocity of CoM of the generated gait.

6. Discussion

6.1. Design parameters
In our framework, there are some design parameters:

- learning algorithm parameters
  $K_{(\cdot)}$, $K_{\rho(\cdot)}$ and $\epsilon_{e(\cdot)}$
- weighting matrices and coefficients of the cost function
  Λ_y, $\Lambda_{\dot{y}}$, $\Lambda_{\bar{u}}$, γ_{u_ρ} and γ_{y_ρ}

First of all, each of them has to be positive definite matrix or positive number. We should choose $\epsilon_{e(\cdot)}$ values small enough so that the approximation in Eq. (31) holds. We let all $\epsilon_{e(\cdot)}$ be a constant in both simulations.

The parameters $K_{(\cdot)}$ and $K_{\rho(\cdot)}$ are the step parameters in the steepest descent method. These parameters have an effect on the convergence speed of the algorithm and have little effect on the generated gait. They also compensate the increase of the cost function due to the small variation of the initial condition of each walking cycle, which will be mentioned in Section 6.3. Although, in general, the step parameters in the steepest descent method are decided by the line search method, we cannot use it, because the plant model is not available. We let all $K_{(\cdot)}$ and $K_{\rho(\cdot)}$ values be constants in both simulations, respectively. However, an appropriate convergent sequence may be effective such that their elements are large in the beginning of learning and they gradually become smaller.

Besides, weighting matrices and coefficients have an effect on the generated gait. To prioritize making the configuration coordinate q periodic, we choose the coefficients of Λ_y larger than the other weighting coefficients. The reason why the coefficients of $\Lambda_{\bar{u}}$ are chosen much smaller than the others is that the cost with respect to the input is evaluated relatively much bigger than those with respect to the output and its time derivative constraints because of the filter function ν_1.

Since $\Lambda_{\dot{y}}$ has an effect on the velocity of the generated gait and γ_{u_ρ} and γ_{y_ρ} have an effect on the convergence speed of the virtual constraint, we should set them according to their priorities. Since we prioritize comparing the velocity with different references, we let $\Lambda_{\dot{y}}$ bigger than γ_{u_ρ} and γ_{y_ρ} in the simulations. Although we do not prioritize the convergence of k_c to 0 in those cases, we observe that k_c converges to 0 by letting γ_{u_ρ} and γ_{y_ρ} large in other simulations.

Let us note that since the proposed method is based on the steepest descent method, achieving only a local minimum is guaranteed. Hence, there is no guarantee that the resultant trajectory corresponding to the local minimum is always admissible walking pattern. Unless the resultant optimal trajectory is admissible, the learning procedure has to be executed again with different initial condition or design parameters.

6.2. Computational cost
The proposed algorithm does not have time-consuming calculations such as large-scale inverse matrices nor calculations of dynamics. Furthermore, it does not require enormous amount of memory. Let us see Eqs. (46) and (48) in the proposed algorithm. First, the time-reversal operator $\mathcal{R}(\cdot)$ can be calculated by just sorting an argument vector backward. In the $(2i+1)$th iteration, the calculation of the differential operator $\mathcal{D}_t(\cdot)$ can be obtained by a numerical differentiation algorithm, and the adjoint calculation $(\delta F_v(\dot{y}_{(2i)} - v_{\text{ref}}))^*$ is easily obtained by Eq. (42). Since a single update of the proposed learning requires only two experimental data, that is, $2i$th and $(2i+1)$th input/output data are required for the ith iteration. Since the other previous data are not required, enormous amount of memory is not necessary.

6.3. Consideration of the variation of the initial condition
Although the virtual constraint with sufficiently large constraint parameter k_c makes the trajectory symmetric, the initial conditions for each walking step may differ slightly. We can evaluate how the mismatch error of the initial condition affects the cost function, and can reflect the evaluation to choosing the step parameters $K_{(\cdot)}$ and $K_{\rho(\cdot)}$ in order to compensate the increase of the cost function. In order to deal with the variation of the initial condition, we consider the plant system as $\Sigma : X \times U \to X \times Y :$ $(x_{t^0}, u) \mapsto (x_{t^1}, y)$:

$$\begin{cases} \dot{x} = (J(x) - R(x)) \dfrac{\partial H(x, u)^\top}{\partial x}, \ x(t^0) = x_{t^0}, \\ y = -\dfrac{\partial H(x, u)^\top}{\partial u}, \\ x_{t^1} = x(t^1), \end{cases}$$

We calculate the Fréchet derivative of the cost function considering the effects of the variations of x_{t^0} and x_{t^1} as (see Eq. (29) for comparison):

$$\begin{aligned}
&\delta\hat{\Gamma}_e(u_e, y_e,)(\delta x_{t^0}, \delta u_e, \delta x_{t^1}, \delta y_e) \\
&= \langle \nabla_{u_e}\hat{\Gamma}_e + \pi_{U_e} \circ (\delta\Sigma_e(x_{t^0}, u_e))^*(\nabla_{x_{t^1}}\hat{\Gamma}_e, \nabla_{y_e}\hat{\Gamma}_e), \delta u_e \rangle_{U_e} \\
&\quad + \langle \nabla_{x_{t^0}}\hat{\Gamma}_e + \pi_X \circ (\delta\Sigma_e(x_{t^0}, u_e))^*(\nabla_{x_{t^1}}\hat{\Gamma}_e, \nabla_{y_e}\hat{\Gamma}_e), \delta x_{t^0} \rangle_X \\
&= \langle \nabla_{u_e}\hat{\Gamma}_e + \pi_{U_e} \circ (\delta\Sigma_e(x_{t^0}, u_e))^*(0, \nabla_{y_e}\hat{\Gamma}_e), \delta u_e \rangle_{U_e} \\
&\quad + \langle \pi_X \circ (\delta\Sigma_e(x_{t^0}, u_e))^*(0, \nabla_{y_e}\hat{\Gamma}_e), \delta x_{t^0} \rangle_X,
\end{aligned}$$

where $\pi_{(\cdot)}$ represents the projection mapping onto $(\cdot)$ and $\circ$ denotes the composition. Since the cost function is not a functional with respect to x_{t^0} nor x_{t^1}, $\nabla_{x_{t^0}}\hat{\Gamma}_e = \nabla_{x_{t^1}}\hat{\Gamma}_e = 0$.

Then, the variation of the cost function with respect to that of the initial condition δx_{t^0} is given by

$$\langle \pi_X \circ (\delta \Sigma_e(x_{t^0}, u_e))^*(0, \nabla_{y_e} \hat{\Gamma}_e), \delta x_{t^0} \rangle_X. \qquad (50)$$

According to refs. [21, 26], the variational symmetry gives the following relationship around a symmetric trajectory (which is an extension of Eq. (10), see also Remark 1):

$$(\delta \Sigma_e(x_{t^0}, u_e))^* = \mathcal{S}^{-1} \circ \delta \Sigma_e(x_{t^0}, u_e) \circ \mathcal{S}, \qquad (51)$$

where the operator $\mathcal{S} : X \times U_e \to X \times U_e$ is defined by

$$\mathcal{S}(x_{t^0}, u_e) := (-(J - R)T x_{t^0}, \mathcal{R}(u_e)),$$

where note that T is given in Remark 3 as $T = \mathrm{diag}(I, -I)$. From Eq. (51), we have

$$
\begin{aligned}
&\pi_X \circ (\delta \Sigma_e(x_{t^0}, u_e))^*(0, \nabla_{y_e} \hat{\Gamma}_e) \\
&= \pi_X \circ \mathcal{S}^{-1} \circ \delta \Sigma_e(x_{t^0}, u_e) \circ \mathcal{S}(0, \nabla_{y_e} \hat{\Gamma}_e) \\
&= \pi_X \circ (-T^{-1}(J - R)^{-1} \delta x_{t^1}, \mathcal{R} \circ \delta \Sigma_e^{x_{t^0}}(u_e)(\mathcal{R}(\nabla_{y_e} \hat{\Gamma}_e)) \\
&= -T^{-1}(J - R)^{-1} \delta x_{t^1}.
\end{aligned}
$$

From Eq. (50), the variation of the cost function with respect to δx_{t^0} is given by $-T^{-1}(J - R)^{-1} \delta x_{t^1}^\top \delta x_{t^0}$. Since J and R are given by the form of (27), and the dissipation matrix K_D in R is constant, we can evaluate an increase of the cost function of the next $j + 1$th iteration due to the mismatch error of the jth initial condition $x_{t^0, j}$ by only information of $\delta x_{t^0, j} := x_{t^0, j} - x_{t^0, j-1}$ and $\delta x_{t^1, j} := x_{t^1, j} - x_{t^1, j-1}$.

Therefore, we can compensate this increase of the cost function by choosing the $j + 1$th learning gain $\mathrm{diag}\{K_{(j+1)}, K_{\rho(j+1)}\}$ so that $\|\mathrm{diag}\{K_{(j+1)}, K_{\rho(j+1)}\}\| > (x_{t^1, j} - x_{t^1, j-1})^\top (x_{t^0, j} - x_{t^0, j-1}) \|K_D\|$. Then, it is guaranteed that the cost function decreases even if the mismatch of the initial conditions exists.

6.4. Future works
Uncertainties, for example measurement noise and perturbations from environmental disturbances during learning, may cause problems in practice. To solve these problems, we consider the plant system with the above uncertainties as a stochastic system and focus on stochastic control theory to take into account disturbances during experiments. In ref. [36], we have extended the deterministic Hamiltonian systems (1) to stochastic Hamiltonian ones, whose dynamics are described by stochastic differential equations. Furthermore, since the variational symmetry of a deterministic Hamiltonian system plays an important role in our learning method, we have investigated a corresponding symmetric property of the stochastic Hamiltonian system in ref. [37]. Now we are tackling extension of the proposed learning framework to stochastic Hamiltonian systems. The relation with other stochastic optimal control methods, e.g. refs. [17, 18], will be interesting.

Besides, application of this method to the robots with multiple DoF is a very important issue. For now, we consider a 12-DoF 3D biped robot and just apply the ILC part

of the proposed method to the robot, that is, we repeat initializations and experiments. We start learning from a pre-designed walking trajectory and try to improve the initial trajectory in order to investigate applicability of the proposed framework. We have already obtained some results of trajectory optimization, and we expect that it is applicable to the complicated robots. Experimental evaluation is also important. We have already developed a 5-DoF planar biped robot as a testbed and are preparing experiments.

Since the proposed method is classified as feedforward control, an orbitally stabilizer, e.g. trajectory tracking control, should be applied to the generated gait after learning in order to render the gait stable. We have already proposed a stochastic trajectory tracking control method in ref. [38], which is applicable even in the presence of noise and would be useful for the purpose.

Those works will be reported in future publications.

7. Conclusion
In this paper, we have proposed a repetitive control type optimal gait generation framework by unifying learning control and parameter tuning. The proposed learning optimal control method of Hamiltonian systems unifying ILC and IFT plays an important role in our framework. It allows one to simultaneously obtain an optimal feedforward input and a tuning parameter for a plant system, which is an optimal solution to infinite-dimensional optimal control problem without any finite-dimensional approximation. The symmetric property of Hamiltonian systems allows one to directly provide the gradient of the cost function without the precise knowledge of the plan model. The feature of the proposed method is that the robot keeps on walking and improves the walking gait due to virtual constraints by a potential energy. That is, this method does not need to repeat experiments under the same initial condition, which is necessary for conventional ILC frameworks. The proposed technique also differs from some conventional methods using virtual constraints in that it automatically mitigates the strength of the constraints by IFT according to the progress of learning by ILC. Finally, numerical simulations demonstrate the effectiveness of the proposed method.

References
1. M. Vukobratović and J. Stepanenko, "On the stability of anthropomorphic systems," *Math. Biosci.* **15**, 1–37 (1972).
2. A. Takanishi, M. Ishida, Y. Yamazaki and I. Kato, "The Realization of Dynamic Walking by the Biped Walking Robot WL-10RD," **In**: *Proceedings of the International Conference on Advanced Robotics* (1985) pp. 459–466.
3. K. Hirai, M. Hirose, Y. Haikawa and T. Takenaka, "The Development of Honda Humanoid Robot," **In**: *Proceedings*

of the IEEE International Conference on Robotics and Automation (1998) pp. 1321–1326.

4. S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi and H. Hirukawa, "Biped Walking Pattern Generation by Using Preview Control of Zero-Moment Point," **In**: *Proceedings of the IEEE International Conference on Robotics and Automation* (2003) pp. 1620–1626.

5. T. McGeer, "Passive dynamic walking," *Int. J. Robot. Res.* **9**(2), 62–82 (1990).

6. M. Garcia, A. Chatterjee, A. Ruina and M. Coleman, "The simplest walking model: Stability, complexity, and scaling," *ASME J. Biomech. Eng.* **120**, 281–288 (1998).

7. K. Osuka and K. Kirihara, "Motion Analysis and Experiments of Passive Walking Robot QUARTET II," **In**: *Proceedings of the 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 IEEE International Conference on Robotics and Automation* (2003) pp. 2478–2483.

9. A. Goswami, B. Espiau and A. Keramane, "Limit cycles in a passive compass gait biped and passivity-mimicking control laws," *Auton. Robots* **4**(3), 273–286 (1997).

10. M. W. Spong, "Passivity Based 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. Robot. Autom.* **20**(3), 565–573 (2004).

12. J. W. Grizzle, G. Abba and F. Plestan, "Asymptotically stable walking for biped robots: Analysis via systems with impulse effects," *IEEE Trans. Autom. Control* **46**(1), 51–64 (2001).

13. S. Hyon and T. Emura, "Symmetric Walking Control: Invariance and Global Stability," **In**: *Proceedings of the IEEE ICRA* (2005) pp. 1455–1462.

14. G. Endo, J. Morimoto, T. Matsubara, J. Nakanishi and G. Cheng, "Learning CPG-based biped locomotion with a policy gradient method: Application to a humanoid robot," *Int. J. Robot. Res.* **27**(2), 213–228 (2008).

15. J. Morimoto and C. Atkeson, "Robust Low Torque Biped Walking Using Differential Dynamic Programming with a Minimax Criterion," **In**: *Proceedings of the 5th International Conference on Climbing and Walking Robots* (2002) pp. 453–459.

16. R. Tedrake, T. W. Zhang and H. S. Seung, "Stochastic Policy Gradient Reinforcement Learning on a Simple 3D Biped," **In**: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (2004) pp. 2849–2854.

17. E. Theodorou, J. Buchli and S. Schaal, "A generalized path integral control approach to reinforcement learning," *J. Mach. Learn. Res.* **11**, 3153–3197 (2010).

18. H. J. Kappen, "An Introduction to Stochastic Control Theory, Path Integrals and Reinforcement Learning," **In**: *Proceedings of the 9th Granada Seminar on Computational Physics: Cooperative Behavior in Neural Systems* (2007) pp. 149–181.

19. S. Satoh, K. Fujimoto and S. Hyon, "Biped Gait Generation via Iterative Learning Control Including Discrete State Transitions," **In**: *Proceedings of the 17th IFAC World Congress* (2008), pp. 1729–1734.

20. S. Satoh, K. Fujimoto and S. Hyon, "A gait generation framework via learning optimal control considering discontinuous state transitions," [in Japanese] *J. Robot. Soc. Japan* **29**(2), 90–100 (2011).

21. S. Satoh, "Control of Deterministic and Stochastic Hamiltonian Systems Application to Optimal Gait Generation for Walking Robots," *Ph.D. dissertation* (Nagoya University, Aichi, Japan, 2010), available at: http://home.hiroshima-u.ac.jp/satoh/index.html.

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

23. P. E. Crouch and A. J. van der Schaft, *Variational and Hamiltonian Control Systems*. Lecture Notes on Control and Information Science, vol. 101 (Springer, Berlin, 1987).

24. B. Maschke and A. J. van der Schaft, "Port-Controlled Hamiltonian Systems: Modelling Origins and System Theoretic Properties," **In**: *Proceedings of the 2nd IFAC Symposium on Nonlinear Control Systems* (1992) pp. 282–288.

25. S. Arimoto, S. Kawamura and F. Miyazaki, "Bettering operation of robotics," *J. Robot. Syst.* **1**(2), 123–140 (1984).

26. K. Fujimoto and I. Koyama, "Iterative Feedback Tuning for Hamiltonian Systems," **In**: *Proceedings of the 17th IFAC World Congress* (2008) pp. 15,678–15,683.

27. F. D. Bruyne, B. D. O. Anderson, M. Gevers and N. Linard, "Iterative Controller Optimization for Nonlinear Systems," **In**: *Proceedings of the 36th IEEE Conference on Decision and Control*, vol. 4 (1997), pp. 3749–3754.

28. H. Hjalmarsson, "Iterative feedback tuning – An overview," *Int. J. Adapt. Control. Signal Process.* **16**, 373–395 (2002).

29. S. Hara, Y. Yamamoto, T. Omata and M. Nakano, "Repetitive control system: A new type servo system for periodic exogenous signals," *IEEE Trans. Autom. Control* **33**(7) 659–668 (1988).

30. 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.

31. S. Satoh, K. Fujimoto and S. Hyon, "Periodic Gait Generation via Repetitive Optimal Control of Hamiltonian Systems," **In**: *Proceedings of the 18th IFAC World Congress* (2011) pp. 6912–6917.

32. K. Fujimoto and T. Sugie, "Canonical transformation and stabilization of generalized Hamiltonian systems," *Syst. Control Lett.* **42**(3), 217–227 (2001).

33. P. Gregorio, M. Ahmadi and M. Buehler, "Design, control, and energetics of an electrically actuated legged robot," *IEEE Trans. Syst. Man Cybern.* **27**(4), 626–634 (1997).

34. F. Asano, Z. W. Luo and M. Yamakita, "Biped gait generation and control based on a unified property of passive dynamic walking," *IEEE Trans. Robot.* **21**(4), 754–762 (2005).

35. S. Collins, A. Ruina, R. Tedrake and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science*, **307**(5712), 1082–1085 (2005).

36. S. Satoh and K. Fujimoto, "On Passivity Based Control of Stochastic Port-Hamiltonian Systems," **In**: *Proceedings of the 47th IEEE Conference on Decision and Control* (2008) pp. 4951–4956.

37. S. Satoh and K. Fujimoto, "A Symmetric Structure of Variational and Adjoint Systems of Stochastic Hamiltonian Systems," **In**: *Proceedings of the 49th IEEE Conference on Decision and Control* (2010) pp. 1423–1428.

38. S. Satoh and K. Fujimoto, "Observer Based Stochastic Trajectory Tracking Control of Mechanical Systems," **In**: *Proceedings of the ICROS–SICE International Joint Conference* (2009) pp. 1244–1248.