Bayesian Inference for Path Following Control of Port-Hamiltonian Systems with Training Trajectory Data

Yuki OKURA *, Kenji FUJIMOTO **, Ichiro MARUTA **, Akio SAITO ***, and Hidetoshi IKEDA ***

Abstract: This paper describes a procedure to design a path following controller of port-Hamiltonian systems based on a training trajectory dataset. The trajectories are generated by human operations, and the training data consist of several trajectories with variations. Hence, we regard the trajectory as a stochastic process model. Then we design a deterministic controller for path following control from the model. In order to obtain reasonable design parameters for a path following controller from the training data, Bayesian inference is adopted in this paper. By using Bayesian inference, we estimate a probability density function of the desired trajectory. Moreover, not only the mean value of the trajectory but also the covariance matrix is acquired. A potential function for path following control is obtained from the probability density function. By incorporating the covariance information into the control system design, it is possible to create a potential function that takes into account uncertainty at each position on the trajectory, and it is expected to construct a control system that generates appropriate assist force for a human operator.

Key Words: path following control, Bayesian inference, port-Hamiltonian systems.

1. Introduction

The progress of the declining birthrate and aging society is one of the big social problems, and it is possibly necessary to cope with the problem of a serious shortage of manpower. In this paper, as one of the solutions to the problem, controller design for good human-machine cooperation is considered.

Let us imagine a control system which physically assists human operation of moving an object from the start point to the end point. In this situation, it is assumed that the trajectory of the object generated by human operation, particularly that corresponding to middle points between the start point and the end point, varies by the individual human operator. Therefore, it is desirable to design a control system that allows some variation caused by the different operation of individual operators. In other words, it is desirable to design a control system that does not forcibly reduce variations due to individual operations by adjusting the assist force from the mechanical system in the directions orthogonal to the desired trajectory.

As control methods to make the system track the desired path, trajectory tracking control [1]–[4] and path following control [5]–[15] are proposed. In these control methods, a virtual path, trajectory tracking control [1]–[4] and path following control [5]–[15] are designed to achieve the objective, and by changing the gradient of the potential function, the magnitude of the control input of the mechanical system can be adjusted. On the other hand, when humans and mechanical systems interact physically, it is important to adjust the magnitude of the control input of the system, that is, to adjust the gradient of the potential function. How to select the potential function at the position on the path is left to the designer.

In order to tackle this problem, our group proposed a procedure to design a path following controller from a training trajectory dataset [16]. In [16], in order to obtain the design parameters for path following control, the desired momentum is modeled as the function of position. However, the trackability of the desired path strongly depends on the choice of the estimation model of the desired momentum.

In this paper, by modifying the method in [16], we propose a path following control of a port-Hamiltonian system from the training trajectory data of the system obtained in the past operations. The training data consists of several trajectories generated by human operators. Although they aim to achieve the same task, they are slightly different from each other. Hence, we regard the trajectory as a stochastic process model. Then we design a deterministic controller for path following control from the model.

In order to obtain reasonable design parameters for the path following controller from the training data, Bayesian inference [17] is utilized. By Bayesian inference, not only the mean but also the covariance of the desired trajectory can be estimated, and the covariance information express the degree of variation caused by the difference among individual operations. Therefore, by incorporating the covariance information into the design parameters of the controller, it is possible to create a potential function that takes into consideration the variation on the trajectories, and it is expected to construct a control system that generates appropriate assist force for a human operator.

The organization of this paper is as follows. First of all, path following control of port-Hamiltonian systems is briefly outlined. Next, interpretation of path following control by the
probability density function estimated from the training trajectory data is explained. Then we propose a method of constructing a potential function from the probability density function. Application of the proposed method using Bayesian inference obtains a potential function for path following control from past trajectory training data. Finally, as a simple experiment, path following controller design from a trajectory dataset of mouse dragging by an operator is considered, and this experiment illustrates the effectiveness of the proposed method.

Some notations used in this paper are defined as follows.

- The Kronecker product of matrices $A$ and $B$ is denoted by $A \otimes B$.
- The vec operator with respect to a matrix $A = (a_1, \ldots, a_n) \in \mathbb{R}^{m \times n}$ is denoted by
  $$\text{vec}(A) = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix}.$$
- The Gaussian distribution of a vector valued variable $x \in \mathbb{R}^n$ with a mean $\mu \in \mathbb{R}^n$ and a covariance $\Sigma \in \mathbb{R}^{n \times n}$ is denoted by
  $$\mathcal{N}(x|\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n|\Sigma|}} \exp \left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right).$$

2. Preliminaries

In this section, the existing results on controller design for path following control of the port-Hamiltonian system [15] are briefly summarized.

2.1 Path Following Control of Port-Hamiltonian Systems

In this paper, consider port-Hamiltonian systems with the following form.

$$\begin{cases} \dot{q} = \begin{pmatrix} 0 \\ -J_{12} \end{pmatrix} + \begin{pmatrix} \frac{\partial H}{\partial p} \\ \frac{\partial G}{\partial p} \end{pmatrix}, \\ H(q, p) = \frac{1}{2} p^T M(q)^{-1} p, \end{cases}$$

where $q \in \mathbb{R}^l$ is the position, $p \in \mathbb{R}^m$ is the momentum, $x = (q^T, p^T)^T \in \mathbb{R}^{l+m}$ is the state, $u \in \mathbb{R}^n$ is input, $J_{12} \in \mathbb{R}^{l \times m}$, $J_{22} \in \mathbb{R}^{m \times m}$ is a skew-symmetric matrix, $R \in \mathbb{R}^{m \times m}$ is a symmetric positive semi-definite matrix, and $M(q) \in \mathbb{R}^{m \times m}$ and $G(q) \in \mathbb{R}^{l \times m}$ are nonsingular matrices. Fully actuated mechanical systems, systems with a class of nonholonomic constraints, and dynamics of LC-circuits can be expressed in this form [18]–[20].

In order to achieve path following control, we need to determine the following design parameters.

- A desired trajectory of position which does not branch and has no intersection.
- Desired momentum $p_{\text{ref}}(q)$ in order for the system to move along the desired path.
- A time invariant potential function $U(q)$ which takes its minimum value 0 on the desired path.

The strategy is to design the feedback controller $u = \beta_p$ by using generalized canonical transformations [21]. Based on the passivity based approach, the following transformed Hamiltonian function $\tilde{H}$ for path following control is obtained by adding an artificial potential function to the Hamiltonian $H$ of the original system (1).

$$\tilde{H} = H - p^T M(q)^{-1} p_{\text{ref}}(q) + \frac{1}{2} \beta_p^T M(q)^{-1} \beta_p + U(q)$$

$$= \frac{1}{2} (p - p_{\text{ref}}(q))^T M(q)^{-1} (p - p_{\text{ref}}(q)) + U(q).$$

The first term in the right hand side $(1/2)(p - p_{\text{ref}}(q))^T M(q)^{-1} (p - p_{\text{ref}}(q))$ represents the error kinetic energy, and the second term $U(q)$ represents the potential energy to make the trajectory converge to the desired path. A feedback controller $\beta_p$ is obtained as follows:

$$\beta_p = -G^T \begin{cases} -M^T \frac{\partial M}{\partial q} J_{12} M^{-1} p \\ -J_{12} \frac{\partial M}{\partial q} M^{-1} - J_{12} \frac{\partial M}{\partial q} M^{-1} p^T M^{-1} \\ + M^T \frac{\partial M}{\partial p} J_{12} M^{-1} p + J_{12} \frac{\partial U}{\partial q} \end{cases} \right) \right) \beta_p$$

$$- KG^T M^{-1} (p - p_{\text{ref}}).$$

where $K(q, p) > 0 \in \mathbb{R}^{m \times m}$.

Therefore, when an appropriate potential function $U(q)$ and desired momentum $p_{\text{ref}}(q)$ are obtained, path following control is achieved by the feedback controller (4). In the next section, we try to find $U(q)$ and $p_{\text{ref}}(q)$ for path following control via Bayesian inference.

3. Main Results

This section discusses how to design a time invariant potential function $U(q)$ and a desired momentum $p_{\text{ref}}(q)$ for path following control of port-Hamiltonian systems from training trajectory data. First, a probability density function of the trajectory and a potential function are compared, and a strategy for selecting a potential function from a probability density function of the trajectory is described.

After that, the potential function is designed using the probability density function of trajectory estimated by Bayesian inference, and the property of the potential function is shown. Then, how to construct the desired momentum function $p_{\text{ref}}(q)$ is explained.

3.1 Problem Statement

The problem statement is briefly explained here. In this paper, we make the following assumption.

- Human operators generate several trajectories for the same task to move an object from a given starting position to a desired terminal one. We call a collection of such trajectories as a training dataset.
The goal of this paper is to obtain the design parameters $U(q)$, $p_{\text{pdf}}(q)$ of a controller $\beta_\text{p}$ in the form (4) from the training trajectory dataset in order to achieve a path following control law for the port-Hamiltonian system in (1), which generates assist force inversely proportional to the fluctuation of the trajectories. The reason for choosing this control law is as follows: Precise position control with high gain feedback is necessary around the point where the variation of the trajectories is small. On the contrary, near the point where the variation of the trajectories is large, the control which does not affect human operation is suitable for assist force.

3.2 Interpretation of a Potential Function of Path Following Control by Probability Density Function

Let us interpret a potential function of path following control from a viewpoint of probability density function of the trajectory here. Suppose that the probability density function of the trajectory $p(x(t))$ is estimated, then the desired trajectory $x_d(t)$ is defined by the maximum likelihood value of the probability density function written as follows:

$$x_d(t) = \text{argmax}_x p(x(t)).$$

(5)

In order to design a potential function, we want to evaluate the closeness of each point $x$ to the desired path. Therefore, the likelihood of closeness of each point $x$ to the desired path is defined as follows:

$$L(x) = \max_t p(x(t)).$$

(6)

The closer a point $x$ is to the desired path, the larger the value of $L(x)$ is.

In order for the state $x$ to track an arbitrary point on the desired path $x_{\text{path}}(t) \in \mathbb{R}^n$, $t \in [0, T]$, we need to design a potential function $V$ which takes its minimum $V(x_{\text{path}}) = 0$ and $V(x) > 0, x \neq x_{\text{path}}$. The closer a point $x$ is to the desired path, the smaller the value of $V(x)$ is. Thus, from the relationship between $L(x)$ and $V(x)$, a candidates of a potential function is the minus maximum logarithm likelihood $L(x)$.

$$V_m(x) = -\log L(x) = -\max_t p(x(t)).$$

(7)

Therefore, when we adopt $V_m(x)$ as a potential function and design a controller in order to reduce the value $V_m(x)$, the likelihood of closeness of each point $x$ to the desired path is maximized, and $V_m(x)$ becomes a statistically meaningful value.

However, depending on the shape of $p(x(t))$, the minimum value of $V_m(x)$ is sometimes not equal to 0 on the desired path. In order for the potential function to take minimum value 0 on the path, a modified potential function $V(x)$ is calculated as follows:

$$V(x) = k(V_m(x) - V_0(x)), $$

(8)

where $V_0$ is a design parameter satisfying

$$V_m(x_{\text{path}}(t)) - V_0(x_{\text{path}}(t)) = V(x_{\text{path}}(t)) = 0 \ \forall t \in [0, T],$$

(9)

and $k > 0 \in \mathbb{R}$ is a coefficient to adjust the magnitude of the potential function.

3.3 Design of Potential Function $U(q)$ by Bayesian Inference

The procedure of acquiring the potential function from a probability density function of the trajectory is described before. Hence, when a probability density function of the desired trajectory can be calculated from a training trajectory data set, a potential function for path following control is obtained. However, in general, it is difficult to obtain the density function with an arbitrary shape. In this paper, we focus on the probability density function expressed by a Gaussian distribution. Here the procedure to obtain the probability density function from the training trajectory data by Bayesian inference is explained.

The trajectory is modeled as follows:

$$q(t) = F\phi(t) + w(t),$$

$$= \text{vec}(F)^T(\phi(t) \otimes I)^T + w(t),$$

(10)

where $t \in [0, 1]$ is the normalized time parameter, $I$ is the identity matrix, $\phi(t) = (\phi_1(t), ..., \phi_M(t)) \in \mathbb{R}^M$ is the basis function, $F \in \mathbb{R}^{N \times M}$ is a weighting matrix, and $w$ is the noise dependent on time $t$ defined as

$$w(t) \in \mathbb{R}^l, \ p(w) = N(w|0, Q(t)),$$

where $Q(t) \in \mathbb{R}^{l \times l}$ is a positive definite matrix. As a training dataset, $N$ pairs of position and time $(q^i, t^i), i = 1, ..., N$ are acquired. For simplicity of notation, we define $q_{\text{train}} := (q^1, ..., q^N)$ and $t_{\text{train}} := (t^1, ..., t^N)$.

In order to design a potential function, we need to obtain a predictive distribution $p(q|q_{\text{train}}, t_{\text{train}})$. The predictive distribution is calculated by

$$p(q|q_{\text{train}}, t_{\text{train}}) = \int p(q|F\phi(t))p(\text{vec}(F)|q_{\text{train}}, t_{\text{train}})dF,$$

(11)

and $p(\text{vec}(F)|q_{\text{train}}, t_{\text{train}})$ is

$$p(\text{vec}(F)|q_{\text{train}}, t_{\text{train}}) \propto p(q_{\text{train}}, t_{\text{train}}|\text{vec}(F))p(\text{vec}(F)).$$

Thus, we have to calculate $p(q_{\text{train}}, t_{\text{train}}|\text{vec}(F))$ and $p(\text{vec}(F))$. As a prior distribution, by defining a mean $m$ and a covariance $R$ of $\text{vec}(F)$ properly, $p(\text{vec}(F))$ is written as follows:

$$p(\text{vec}(F)) \sim \text{N}(\text{vec}(F)|m, R),$$

$$\propto \exp\left[-\frac{1}{2}(\text{vec}(F) - m)^T R^{-1}(\text{vec}(F) - m)\right].$$

(12)

On the other hand, $p(q_{\text{train}}, t_{\text{train}}|\text{vec}(F))$ is

$$p(q_{\text{train}}, t_{\text{train}}|\text{vec}(F))$$

$$\sim \prod_{i=1}^{N} N(q^i|F\phi(t^i), Q(t^i))$$

$$\propto \exp\left\{-\sum_{i=1}^{N}(q^i - F\phi(t^i))^T Q^{-1}(q^i - F\phi(t^i))\right\}$$

$$= \exp\left\{-\sum_{i=1}^{N}\left((f_1\phi_1 + f_2\phi_2 + \cdots + f_M\phi_M) - q^i\right)^T \cdot Q^{-1}(f_1\phi_1 + f_2\phi_2 + \cdots + f_M\phi_M) - q^i\right\}.$$
where \( p \) allows:

\[
\Sigma_q = \mathbb{E}[(\hat{q} - \mu_q)^2] = \mathbb{E}[(q - \mu_q)^2]
\]

In order for the potential function to take its minimum value 0 on the desired path and to keep the information of the covariance \( \Sigma_q \) as its Hessian matrix, the function \( V_0(q) \) is designed as

\[
V_0(q) = \max_{t \in [0,1]} \log \left( \frac{p(q(t))}{\max_{t \in [0,1]} p(q(t))} \right) - \max_{t \in [0,1]} \log p(q(t)).
\]

(22)

By substituting (22) and (18) into (21), the potential function \( U \) is obtained as follows:

\[
U(q) = k \min_{t \in [0,1]} \frac{1}{2} (q - \mu_q(t))^2 \Sigma_q(t)^{-1} (q - \mu_q(t))
\]

(23)

The properties of \( U(q) \) and \( V_0(q) \) are summarized as follows.

**Property 1.** The potential function \( U(q) \) in (23) takes its minimum value 0 on arbitrary point on \( \mu_q(t) \), \( t \in [0,1] \).

**Proof.** Thanks to the positive definiteness of \( Q \), the covariance matrix \( \Sigma_q(t) \) is also positive definite, that is, the function \( V(q,t) = (q - \mu_q(t))^2 \Sigma_q(t)^{-1} (q - \mu_q(t)) > 0 \) \( \forall q - \mu_q(t) \neq 0 \), and this function takes its minimum value 0 with \( q - \mu_q(t) = 0 \) \( \forall t \in [0,1] \). Therefore, \( \min_{t \in [0,1]} V(q,t) \) takes minimum value 0 on \( \mu_q(t) \), \( t \in [0,1] \), and \( U(q) \) in (23) takes 0 on the set \( \mu_q(t), t \in [0,1] \) and \( U(q) > 0, q \neq \mu_q(t) \). \( \square \)

**Property 2.** If \( \Sigma_q(t) \) is constant, then \( V_0 = -\log(1/(2\pi)^N|\Sigma_q|) \), and \( U(q) \) in (23) becomes a quadratic function of \( q \) and its Hessian matrix is equal to \( \Sigma_q \).

Property 1 claims that the potential function \( U(q) \) in (23) takes its minimum value 0 on the mean value of the predictive distribution \( p(q|\hat{q}_{train}, t) \), \( t \in [0,1] \). Moreover, Property 2 implies that the function \( V_0 \) cancels just the offset of \( -\max_{t \in [0,1]} \log p(q(t)) \) and does not modify the quadratic term in \( -\max_{t \in [0,1]} \log p(q(t)) \). Thus, the function (23) keeps the condition for the potential function of path following control in the sense of the mean value of the probability density function without the loss of information of the covariance.

In general, it is difficult to obtain the analytic solution of \( U(q) \) in (23). In a numerical example in the next section, the potential function \( U(q) \) in (23) is obtained by numerical calculation.

### 3.4 Estimation of Desired Momentum

The estimation of the desired momentum is explained here. In [16], the estimation modeled of desired momentum is defined by a function of position. However, this model sometimes causes poor trackability to the desired path. In order to improve the trackability, the model structure of the port-Hamiltonian system and estimation model of normalized time are utilized.

The desired trajectory of the position \( q_d \) is defined by

\[
q_d(t) = \arg \max_{\hat{q}} p(q(t)|\hat{q}_d(t)).
\]

(24)

By using the relationship between \( \dot{q} \) and \( p \), the desired momentum \( p_d(t) \) is written as

\[
p_d(t) = M(q_d(t))\dot{q}_d(t).
\]

(25)
In order to adjust the momentum along to the desired trajectory, the desired momentum $p_{\text{ref}}$ is calculated as

$$p_{\text{ref}} = \frac{p_d(t)}{\|[p_d(t)]||} f(t), \quad (26)$$

where the function $f(t) \in \mathbb{R}$ is the design parameter to adjust the magnitude of the momentum. Thus, when the normalized time $t$ is modeled as a function of $q$, the time invariant desired momentum $p_{\text{ref}}(q)$ can be obtained.

Assume that the desired momentum is a function of the position $q$. The normalized time $t$ is modeled as follows:

$$t(q) = G\psi(q) + v(q). \quad (27)$$

Here $\psi(q) \in \mathbb{R}^d$ is the basis function, $G \in \mathbb{R}^{m \times d}$ is a weighting matrix, $v(q)$ is the noise described as $v(q) \in \mathbb{R}^m$, $p(v) = N(\nu(0), Q_p(q))$, $Q_p \in \mathbb{R}^{m \times m}$.

By using the same procedure as written in Section 3.3, the mean $\mu_q$ and the covariance $\Sigma_q$ of the predictive distribution $p(q(t)|q_{\text{train}}, q_{\text{test}}, q)$ are acquired via Bayesian inference with training dataset $q_{\text{train}} = (q^1, \ldots, q^n)$ and $t_{\text{train}} = (t^1, \ldots, t^n)$.

From the above, the desired momentum $p_{\text{ref}}$ is obtained as follows:

$$p_{\text{ref}}(q) = \frac{p_d(t(q))}{\|[p_d(t(q)]||} f(t(q)). \quad (28)$$

### 3.5 Algorithm

Here we summarize the procedure of parameter design for path following control from training trajectory data as the following algorithm.

**Algorithm**

1. Given: $N$ pairs of training trajectory datasets $(q^i, p^i, t^i)$ $i = 1, \ldots, N$, design parameters $k, f(t)$, the estimation models of trajectory (10) and of normalized time (27).
2. Calculate $\mu_q$ and $\Sigma_q$ according to (19) and (20).
3. Calculate $U(q)$ according to (23).
4. Calculate $p_{\text{ref}}(q)$ according to (28).
5. Calculate a path following controller $\beta_q$ according to (4).

### 4. Experimental Result

This section shows the effectiveness of the proposed method of constructing the potential function and the desired momentum for path following control by a simple experiment.

The experimental setting is illustrated in Fig. 1. Here the actual mouse dragging is treated as one of the human skills. The task setting is illustrated in Fig. 2. One operator drags the mouse from the start point to the goal point, avoiding the obstacles on the display. The objective of this experiment is to obtain the path following controller from the operator’s mouse dragging data.

In this experiment, the mouse pointer is virtually modeled as the following port-Hamiltonian system.

$$\dot{q} = \left[ \begin{array}{c} 0 \\ 0 \\ \dot{p} \end{array} \right] = \left[ \begin{array}{c} \frac{\partial f}{\partial q} \\ \frac{\partial f}{\partial p} \\ 0 \end{array} \right] + \left[ \begin{array}{c} 0 \\ 1 \end{array} \right] u,$$

$$y = \frac{\partial H}{\partial p} (\dot{q}, \dot{p}),$$

$$H = \frac{1}{2} p^T M^{-1} p.$$  

Here $q = (q_1, q_2)^T$, $p = (p_1, p_2)^T$, and $M = I \in \mathbb{R}^{2 \times 2}$.

The training data set shown in Fig. 3 is obtained by four mouse dragging trajectories by one operator. From these training data sets, the potential function $U(q)$ and the desired momentum $p_{\text{ref}}$ are calculated.

The estimation models of the trajectory $q$ in (10) and the normalized time $t$ in (27) are chosen as follows:

\[ q(t) = F\psi(t) + w(t), \quad (30) \]

\[ \psi(t) = \left( \exp \left( \frac{t^2}{2} \right), \exp \left( \frac{(t - 0.2)^2}{18^2} \right), \exp \left( \frac{(t - 0.4)^2}{18^2} \right), \right. \]
\[ \left. \exp \left( \frac{(t - 0.5)^2}{18^2} \right), \exp \left( \frac{(t - 0.7)^2}{18^2} \right), \exp \left( \frac{(t - 1)^2}{9^2} \right) \right)^T, \]

\[ t(q) = G\psi(q) + v(q), \quad (31) \]

\[ \psi(q) = (1, q_1, q_2, q_1^2, q_1 q_2, q_2^2, q_1^3, q_1 q_3, q_2 q_3, q_3^2, q_1^3, q_1 q_3, q_3^2, q_1^3)^T. \]

For these models, the means and the covariances are calculated.
via Bayesian inference. Then, the potential function $U(q)$ in (23) and the desired momentum $p_{\text{ref}}$ in (28) are obtained by numerical computation.

The estimation result of trajectory $q(t)$ is shown in Fig. 4. The gray dots illustrate the position of the training data sets, and the broken line illustrates the estimated trajectory $\hat{q}(t)$. Figure 5 illustrates the time invariant potential function $U(q)$. The arrows in Fig. 6 shows the estimated value of $p_{\text{ref}}(q)$.

By using the potential function $U(q)$ shown in Fig. 5 and $p_{\text{ref}}(q)$ in Fig. 6, the path following controller (4) is acquired. Figure 7 shows the simulation result of path following control. The motion of the system is depicted by the solid line, and the desired path is depicted by the broken line in Fig. 7. Figure 7 illustrates the locus of the position $q$ and shows that the state starting from $q(0) = (0, 0)$ tracks the desired path.

### 5. Conclusions

In this paper, a design method of path following control for port-Hamiltonian systems based on Bayesian estimation using training trajectory data of the past operations has been proposed. Path following control has been interpreted from the viewpoint of the probability density function of the desired trajectory, and a design procedure of a potential function for path following control has been explained. A simple experiment has shown the effectiveness of the proposed method.

Future work will include its extension to cope with paths which have branches or intersections and controller design with other estimation methods.

### References

[1] J.-J.E. Slotine and W. Li: Composite adaptive control of robot manipulators, *Automatica*, Vol. 25, No. 4, pp. 509–519, 1989.
[2] R. Ortega, A. Loria, P.J. Nicklasson, and H. Sira-Ramirez: *Passivity-based Control of Euler-Lagrange Systems: Mechanical, Electrical and Electromechanical Applications*, Springer, 1998.
[3] K. Fujimoto, K. Sakurama, and T. Sugie: Trajectory tracking control of port-controlled Hamiltonian systems via generalized canonical transformations, *Automatica*, Vol. 39, No. 12, pp. 2059–2069, 2003.
[4] K. Fujimoto, K. Sakurama, and T. Sugie: Trajectory tracking control of nonholonomic Hamiltonian systems via generalized canonical transformations, *European Journal of Control*, Vol. 10, No. 5, pp. 421–431, 2004.
[5] J.K. Salisbury: Active stiffness control of a manipulator in cartesian coordinates, *9th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, pp. 95–100, 1980.
[6] N. Hogan: Impedance control: An approach to manipulation, *American Control Conference*, pp. 304–313, 1984.
[7] P.Y. Li and R. Horowitz: Passive velocity field control (pvfc): Part i. geometry and robustness, *IEEE Transactions on Automatic Control*, Vol. 46, No. 9, pp. 1346–1359, 2001.
[8] P.Y. Li and R. Horowitz: Passive velocity field control (pvfc): Part ii. application to contour following, *IEEE Transactions on Automatic Control*, Vol. 46, No. 9, pp. 1360–1371, 2001.
[9] V. Duindam and S. Stramigioli: Port-based asymptotic curve tracking for mechanical systems, *European Journal of Control*, Vol. 10, No. 5, pp. 411–420, 2004.
[10] C. Nielsen and M. Maggiore: Output stabilization and maneuver regulation: A geometric approach, *Systems & Control Letters*, Vol. 55, No. 5, pp. 418–427, 2006.
[11] A. Hladio, C. Nielsen, and D. Wang: Path following for a class of mechanical systems, *IEEE Transactions on Control Systems Technology*, Vol. 21, No. 6, pp. 2380–2390, 2013.
[12] A. Akhtar, C. Nielsen, and S.L. Waslander: Path following using dynamic transverse feedback linearization for car-like robots, *IEEE Transactions on Robotics*, Vol. 31, No. 2, pp. 269–279, 2015.
[13] A.P. Aguiair and J.P. Hespanha: Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty, *IEEE Transactions on Automatic Control*, Vol. 52, No. 8, pp. 1362–1379, 2007.
[14] H. Nakamura: Global nonsmooth control Lyapunov function design for path-following problem via minimum projection method, *IFAC-PapersOnLine*, Vol. 49, No. 18, pp. 600–605, 2016.
[15] Y. Okura and K. Fujimoto: On potential function design for path following control of port-Hamiltonian systems, 2017 *IEEE 56th Conference on Decision and Control (CDC)*,
pp. 2569–2574, 2017.

[16] Y. Okura, K. Fujimoto, A. Saito, and H. Ikeda: On path following control of port-Hamiltonian systems by Bayesian inference with training trajectory data, 2018 IEEE Conference on Decision and Control (CDC), pp. 1956–1960, 2018.

[17] C.M. Bishop: Pattern Recognition and Machine Learning (Information Science and Statistics), Springer, 2006.

[18] A.J. van der Schaft: $L_2$-gain and Passivity Techniques in Nonlinear Control, Springer, 1996.

[19] A.J. van der Schaft and B.M. Maschke: On the Hamiltonian formulation of nonholonomic mechanical systems, Reports on Mathematical Physics, Vol. 34, No. 2, pp. 225–233, 1994.

[20] B.M. Maschke, A.J. van der Schaft, and P.C. Breedveld: An intrinsic Hamiltonian formulation of the dynamics of LC-circuits, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Vol. 42, No. 2, pp. 73–82, 1995.

[21] K. Fujimoto and T. Sugie: Canonical transformation and stabilization of generalized Hamiltonian systems, Systems & Control Letters, Vol. 42, No. 3, pp. 217–227, 2001.

Yuki Okura (Member)

He received his B.S., M.S., and Ph.D. degrees in Engineering from Kyoto University, Japan, in 2014, 2016, and 2019, respectively. He is currently an Assistant Professor at the Department of Electrical and Computer Engineering, Toyama Prefectural University, Japan. His research interests include nonlinear control, optimal control, and control theory for human-machine interaction.

Kenji Fujimoto (Member)

He received the B.Sc. and M.Sc. degrees in Engineering and Ph.D. degree in Informatics from Kyoto University, Japan, in 1994, 1996, and 2001, respectively. He is currently a professor of the Graduate School of Engineering, Kyoto University, Japan. From 1997 to 2004, he was a research associate of the Graduate School of Engineering and the Graduate School of Informatics, Kyoto University, Japan. From 2004 to 2012, he was an associate professor of the Graduate School of Engineering, Nagoya University, Japan. His research interests include nonlinear control and stochastic systems theory.

Ichiro Maruta (Member)

He received the Bachelor of Engineering, Master of Informatics, and Doctor of Informatics degrees from Kyoto University, Kyoto, Japan, in 2006, 2008, and 2011, respectively. He was a research fellow of the Japan Society for the Promotion of Science from 2008 to 2011. From 2012 to 2017, he was an Assistant Professor at the Graduate School of Informatics, Kyoto University. In 2017, he joined the Graduate School of Engineering, Kyoto University, where he is currently a Lecturer of the Department of Aeronautics and Astronautics.

Akio Saito

He received the B.Eng. and M.Eng. degrees in control engineering from the Tokyo Institute of Technology in 2008 and 2010, respectively. Since 2010, he has been with the Mitsubishi Electric Corp. and has been engaged in R&D of industrial control systems, especially industrial servo and robot.

Hidetoshi Ikeda (Member)

He received his B.Eng. and M.Eng. degrees in electrical engineering from Kyoto University in 1989 and 1991, respectively. Since 1991, he has been with the Mitsubishi Electric Corp. and has been engaged in R&D of industrial control systems, especially industrial servo. He is a member of ISCIE, IEEJ, ISIJ, and IEEE.