An Optimal Control Approach for Exploratory Actions in Active Tactile Object Recognition

Daisuke Tanaka, Takamitsu Matsubara, and Kenji Sugimoto

***Abstract*— In this paper, we treat the active tactile object recognition problem using an anthropomorphic robotic hand. Regarding the exploratory action design, to avoid such undesirable situations that the robot might break the object or might get a damage, the compliance of the robot behaviors is important as well as the informativeness of the resulting sensor data. However, most previous studies cannot consider these characteristics simultaneously since they treat the planning problem of exploratory actions separately from the robot control problem. We propose to design the exploratory actions using the formulation of the optimal control problem with the robot dynamics. Our cost function is composed of two terms: the informativeness and the energy consumption that can promote resulting actions to be compliant. The effectiveness of the proposed method is validated for the task of tactile object recognition through physical simulations and experiments in real environment.**

I. INTRODUCTION

The ability of recognizing the situation based on data obtained by such multimodal sensors as vision, auditory and tactile sensors is important for humanoid robots in real environments. For the efficient recognition, it is crucial for the robots to plan and execute clever actions (referred to as *exploration actions*) sequentially so that the resulting sensor data become sufficiently *informative*. In this paper, we treat the tactile object recognition problem using an anthropomorphic robotic hand. More concretely, the robot touches the target object by performing exploratory actions to recognize it based on the obtained tactile data.

Regarding the exploratory action design, to avoid such undesirable situations that the robot might break the object or might get a damage, the compliance of the robot behaviors is important as well as the informativeness of the resulting sensor data. However, most previous studies cannot consider these characteristics simultaneously since they treat the planning problem of exploratory actions separately from the robot control problem (e.g. [1]–[4]).

In this paper, we propose to design the exploratory actions using the formulation of the optimal control problem with the robot dynamics. The optimal control can find a control law that minimizes the resulting cost function. We propose the cost function that is composed of two terms: the informativeness and the energy consumption that can promote resulting actions to be compliant. As a criterion of the informativeness, we adopt the *mutual information*. The effectiveness of the proposed method is validated for the task of tactile object recognition through physical simulations and experiments using an anthropomorphic robotic hand.

The remainder of this paper is organized as follows. Section II describes the procedure of the active object recognition and the definition of the action informativeness. In Section III, our proposed method for the exploratory action design is described. Section IV shows the result of experiments. In Section V, our method is summarized and its future work are also discussed.

*Part of this work was supported by the Tateisi Science and Technology Foundation

All authors are with Graduate School of Information Science, Nara Institute of Science and Technology, Japan {daisuke-t, takam-m, kenji}@is.naist.jp

II. PRELIMINARIES

A. Sequential Active Learning for Object Recognition

We treat the object recognition problem as a parameter estimation problem [3]. We assume that each object has the intrinsic parameter called *object parameter*, and this parameter will be sequentially estimated using the tactile sensor data obtained by the exploratory action.

Generally speaking, the procedure of the active object recognition is summarized as follows:

Step 1: Set an initial guess of the object parameter for the (unknown) target object using a probability distribution (called *object's belief*).
Step 2: Design the optimal exploratory action based on the current object's belief.
Step 3: Obtain the observation (tactile sensor data) by executing the designed action to the target object.
Step 4: Update the object's belief based on the observation.
Step 5: Repeat from Step 2 until the variance of the belief becomes sufficiently small.
Step 6: Determine the result as the object which has the nearest object parameter in the database.

B. Informativeness of Exploratory Action

We use the mutual information [5] as the criterion of informativeness for the exploratory action. The informativeness of exploratory action for each update depends on the current object's belief represented as the probability distribution of an object parameter $\boldsymbol{\theta}$. The mutual information $\mathrm{I}[\boldsymbol{\theta}, \mathbf{y}|\mathbf{x}]$ evaluates the reduced amount of the object parameter's uncertainty when the observation $\mathbf{y}$ is obtained at the state $\mathbf{x}$. In other words, it represents the amount of obtained information.

The mutual information is defined using Kullback-Leibrer

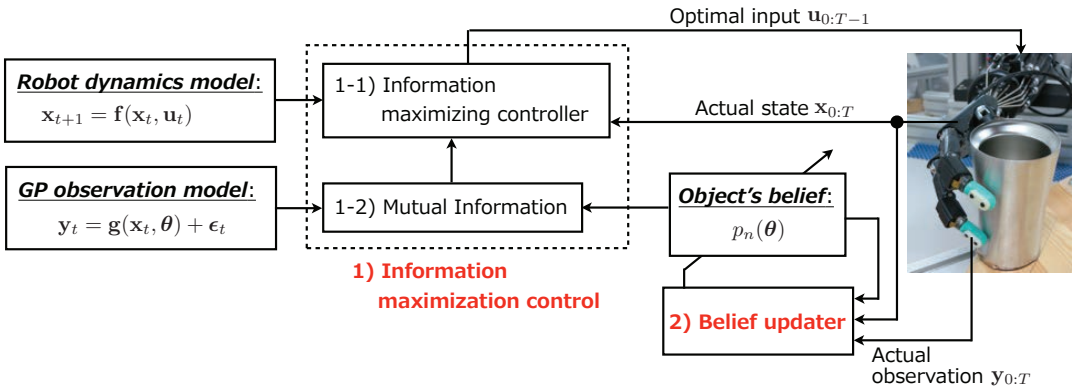


Fig. 1. Overview of the proposed method. 1) Information maximization control consists of the following components: 1-1) Controller that designs informative and compliant actions, 1-2) Mutual information calculation by the observation model constructed with Gaussian process regression, and 2) Object's belief update from obtained observation.

Divergence as follows [6]:

$$\begin{aligned}\mathrm{I}\big[\boldsymbol{\theta}, \mathbf{y}|\mathbf{x}\big] &\triangleq \mathrm{KL}(p(\boldsymbol{\theta}, \mathbf{y}|\mathbf{x}) \| p(\boldsymbol{\theta}) p(\mathbf{y}|\mathbf{x})) \\ &= \iint p(\boldsymbol{\theta}, \mathbf{y}|\mathbf{x}) \log \frac{p(\boldsymbol{\theta}, \mathbf{y}|\mathbf{x})}{p(\boldsymbol{\theta}) p(\mathbf{y}|\mathbf{x})} \mathrm{d}\mathbf{y} \mathrm{d}\boldsymbol{\theta}, \quad (1)\end{aligned}$$

and it is also represented using the entropy $\mathrm{H}[\cdot]$ as follows:

$$\mathrm{I}\big[\boldsymbol{\theta}, \mathbf{y}|\mathbf{x}\big] = \mathrm{H}[\boldsymbol{\theta}] - \mathrm{H}[\boldsymbol{\theta}|\mathbf{y}, \mathbf{x}].$$

We can obtain the effective observation for the parameter estimation by controlling the system to the state sequence that maximizes this quantity.

III. Proposed Method: Optimal Control Approach for Exploratory Actions

A. Overview

The overview of our proposed method for the exploratory action design is shown in Fig. 1. The whole process is roughly divided into the two components. The first component is *information maximization control* that designs the exploratory action considering informativeness and compliance. The design problem is formulated as an optimal control problem (described in Section III.-B) with a mutual information criterion for the informativeness definition. The second component is *belief update* from obtained observation by induced action (described in Section III.-C).

We assume that the robot and tactile sensor for the object recognition are represented as the following the state transition and observation equations:

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t), \quad (2)$$

$$\mathbf{y}_t = \mathbf{g}(\mathbf{x}_t, \boldsymbol{\theta}) + \boldsymbol{\epsilon}_t, \quad (3)$$

where $\mathbf{x} \in \mathbb{R}^{d_\mathbf{x}}$ is the $d_\mathbf{x}$-dimensional (observable) state of the robot, $\mathbf{u} \in \mathbb{R}^{d_\mathbf{u}}$ is the $d_\mathbf{u}$-dimensional input to the robot, $\mathbf{y} \in \mathbb{R}^{d_\mathbf{y}}$ is the $d_\mathbf{y}$-dimensional observation from the robot's sensor, $\boldsymbol{\theta} \in \mathbb{R}^{d_{\boldsymbol{\theta}}}$ is the $d_{\boldsymbol{\theta}}$-dimensional object parameter, and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}})$, $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} = \mathrm{diag}\{\sigma_1^2, \sigma_2^2, \ldots, \sigma_{d_\mathbf{y}}^2\}$ is the $d_\mathbf{y}$-dimensional Gaussian observation noise. We also assume the input of robot is limited to $\mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max}$.

B. Information Maximization Control

We formulate the exploratory action design using the finite horizon optimal control framework [7]. More concretely, Step 2 in the procedure summarized in Section II. can be considered as the problem to find an energy efficient and compliant robot controller that generates a state sequence inducing an informative observation sequence (described in Section III-B-1). However, the informativeness is defined by the mutual information (1) having the computationally intractable double integral form. To simplify the computation, we utilize a Gaussian process observation model [3] as shown in Section III-B-2. Note that the informativeness needs to be maximized as described for exploratory action design, while the optimal control problem is generally formulated as a minimization problem of the cost. The mutual information is converted into a cost to be minimized as shown later.

1) Approximate Optimal Control: We consider the optimal control problem: Find the control law $\mathbf{u}_t = \boldsymbol{\pi}(t, \mathbf{x}_t)$ which minimizes the cost function for the system (2), that is,

$$\begin{aligned}&\underset{\boldsymbol{\pi}}{\text{minimize}} \;\; J_{0:T} \\ &\text{s.t.} \quad \mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t), \quad \mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max}\end{aligned}$$

where

$$J_{0:T} = h(\mathbf{x}_T) + \sum_{t=0}^{T-1} \ell(t, \mathbf{x}, \mathbf{u})$$

is the accumulated cost function, $h(\mathbf{x}_T) \geq 0$ is the terminate cost, and $\ell(t, \mathbf{x}, \mathbf{u}) \geq 0$ represents the running cost. For the exploratory action design, we set the cost function associated by the informativeness and the energy consumption as follows:

$$\ell(t, \mathbf{x}, \mathbf{u}) = q(\mathbf{x}_t) + r(\mathbf{u}_t),$$

where the first term $q(\mathbf{x}_t)$ is related to the informativeness defined by the mutual information $\mathrm{I}\big[\boldsymbol{\theta}, \mathbf{y}|\mathbf{x}\big]$, and the second term $r(\mathbf{u}_t)$ represents the energy consumption.

We utilize the iterative Linear Quadratic Regulator (iLQR, [8]) as a computational efficient and scalable optimal control

solver: the linearized system around the initial state sequence $\bar{\mathbf{x}}_{0:T}$ corresponds to the initial input sequence $\bar{\mathbf{u}}_{0:T-1}$ are constructed, and the local LQR problem is solved for the linearized system. The iLQR also gives a local feedback gains $\mathbf{L}_t$ along $\bar{\mathbf{u}}_{0:T-1}$, therefore, the control law can be given by $\boldsymbol{\pi}(t, \mathbf{x}_t) = \bar{\mathbf{u}}_t + \mathbf{L}_t(\mathbf{x}_t - \bar{\mathbf{x}}_t)$ [8].

2) Mutual Information-based State Cost Function: The observation model (3) is supposed to be modeled using Gaussian Process Regression [9] for each dimension of $\mathbf{y}$: $y_a = g_a(\mathbf{x}, \boldsymbol{\theta}) + \epsilon_a, a \in \{1, 2, \ldots, d_{\mathbf{y}}\}$. For given N-sample training data set $\mathcal{D} = \{\mathbf{x}^{(i)}, \boldsymbol{\theta}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^N$, the predictive distribution of y_a is given as a Gaussian distribution:

$$\begin{aligned} &p(y_a|\mathbf{x}, \boldsymbol{\theta}, \mathbf{X}, \boldsymbol{\Theta}, \mathbf{y}_a) \\ &= \mathcal{N}(\mu_a(\mathbf{x}, \boldsymbol{\theta}; \mathbf{X}, \boldsymbol{\Theta}, \mathbf{y}_a), s_a^2(\mathbf{x}, \boldsymbol{\theta}; \mathbf{X}, \boldsymbol{\Theta}, \mathbf{y}_a)) \end{aligned}$$

where $\mathbf{X}, \boldsymbol{\Theta}$ and $\mathbf{y}_a$ are the training data set corresponding to $\mathbf{x}, \boldsymbol{\theta}$ and y_a respectively. The predictive mean μ_a and variance s_a^2 are given as follows:

$$\begin{aligned} \mu_a(\mathbf{z}; \mathbf{Z}, \mathbf{y}_a) &= \mathbf{k}_a^{\mathrm{T}}(\mathbf{K}_a + \sigma_a^2\mathbf{I})^{-1}\mathbf{y}_a, \\ s_a^2(\mathbf{z}; \mathbf{Z}, \mathbf{y}_a) &= k_a(\mathbf{z}, \mathbf{z}) - \mathbf{k}_a^{\mathrm{T}}(\mathbf{K}_a + \sigma_a^2\mathbf{I})^{-1}\mathbf{k}_a. \end{aligned}$$

Here, $\mathbf{z} = \left[\mathbf{x}^{\mathrm{T}}, \boldsymbol{\theta}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^{d_{\mathbf{z}}}$, $d_{\mathbf{z}} = d_{\mathbf{x}} + d_{\boldsymbol{\theta}}$ is the input vector for the observation model defined for the simplicity, and the vector $\mathbf{k}_a$ is denoted as $\mathbf{k}_a = [k_a(\mathbf{z}^{(1)}, \mathbf{z}), \ldots, k_a(\mathbf{z}^{(N)}, \mathbf{z})]^{\mathrm{T}}$. The matrix $\mathbf{K}_a$ is the kernel matrix with its (i, j)-th element $K_{a,ij} = k_a(\mathbf{z}^{(i)}, \mathbf{z}^{(j)})$. In this paper, the kernel function defined for the calculation of μ_a and s_a^2 is assumed to be the following squared exponential kernel function:

$$k_a(\mathbf{z}, \mathbf{z}') = \alpha_a^2 \exp\Big(-\tfrac{1}{2}(\mathbf{z} - \mathbf{z}')^{\mathrm{T}}\big(\mathbf{H}_a^{\mathbf{z}}\big)^{-1}(\mathbf{z} - \mathbf{z})\Big)$$

where α_a^2 is the variance of g_a. This selection of the kernel function allows us to obtain the Gaussian predictive distribution with approximation in sense of the 1st and 2nd order moments. Here, $\mathbf{x}$ and $\boldsymbol{\theta}$ are assumed to be independent, accordingly $\mathbf{H}_a^{\mathbf{z}}$ is defined as a block diagonal matrix $\mathbf{H}_a^{\mathbf{z}} = \mathrm{block\ diag}\{\mathbf{H}_a^{\mathbf{x}}, \mathbf{H}_a^{\boldsymbol{\theta}}\}$, and $\mathbf{H}_a^{\mathbf{x}}$ and $\mathbf{H}_a^{\boldsymbol{\theta}}$ are diagonal matrices with positive elements. Hyperparameter to be learned is $\gamma_a = \big\{\alpha_a^2, \sigma_a^2, \mathbf{H}_a^{\mathbf{x}}, \mathbf{H}_a^{\boldsymbol{\theta}}\big\}$, and it is optimized by the marginal log-likelihood function [9].

Let us utilize the Gaussian distribution as the object's belief: $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then the joint distribution between $\boldsymbol{\theta}$ and $\mathbf{y}$ given $\mathbf{x}$ is also given by a Gaussian distribution as follows [10]:

$$\begin{aligned} p(\boldsymbol{\theta}, \mathbf{y}|\mathbf{x}) &= \mathcal{N}\left(\left[\begin{array}{c} \boldsymbol{\theta} \\ \mathbf{y} \end{array}\right] \middle| \left[\begin{array}{c} \boldsymbol{\mu} \\ \mathbf{m}(\mathbf{x}) \end{array}\right], \tilde{\boldsymbol{\Sigma}}(\mathbf{x})\right), \\ \tilde{\boldsymbol{\Sigma}}(\mathbf{x}) &= \left[\begin{array}{cc} \boldsymbol{\Sigma} & \mathbf{C}(\mathbf{x}) \\ \mathbf{C}^{\mathrm{T}}(\mathbf{x}) & \boldsymbol{\Phi}(\mathbf{x}, \mathbf{x}) \end{array}\right] \end{aligned}$$

where Φ_{ab} which is the (a, b)-th element of $\boldsymbol{\Phi}(\mathbf{x}, \mathbf{x}') \in \mathbb{R}^{d_{\mathbf{y}} \times d_{\mathbf{y}}}$, and $\mathbf{C}(\mathbf{x}) \in \mathbb{R}^{d_{\boldsymbol{\theta}} \times d_{\mathbf{y}}}$ are defined as follows:

$$\begin{aligned} \Phi_{ab} &= \boldsymbol{\beta}_a^{\mathrm{T}}\boldsymbol{\Lambda}_{ab}(\mathbf{x}, \mathbf{x}')\boldsymbol{\beta}_b - m_a(\mathbf{x})m_b(\mathbf{x}') \\ &\quad + \delta_{\mathbf{x}\mathbf{x}'}\delta_{ab}\Big(\alpha_a^2 - \mathrm{Tr}\big((\mathbf{K}_a + \sigma_a^2\mathbf{I})^{-1}\boldsymbol{\Lambda}_{aa}(\mathbf{x}, \mathbf{x}')\big)\Big), \\ \mathbf{C} &= \boldsymbol{\Psi}(\mathbf{x}) - \boldsymbol{\mu}\mathbf{m}(\mathbf{x})^{\mathrm{T}}. \end{aligned}$$

The a-th entry of $\mathbf{m}(\mathbf{x}) \in \mathbb{R}^{d_{\mathbf{y}}}$ is $m_a = \boldsymbol{\beta}_a^{\mathrm{T}}\boldsymbol{\lambda}_a(\mathbf{x})$, and $\boldsymbol{\beta}_a$ is defined as $\boldsymbol{\beta}_a = (\mathbf{K}_a + \sigma_a^2\mathbf{I})^{-1}\mathbf{y}_a \in \mathbb{R}^N$. Enjoying this result, the double integral in Eq. (1) can be evaluated analytically, and it is represented using the training data and the hyperparameter as follows:

$$\mathrm{I}\big[\boldsymbol{\theta}, \mathbf{y}|\mathbf{x}\big] = -\frac{1}{2}\log\left(\frac{\det\tilde{\boldsymbol{\Sigma}}(\mathbf{x})}{\det\boldsymbol{\Phi}(\mathbf{x}, \mathbf{x})\det\boldsymbol{\Sigma}}\right).$$

See the appendix for the definition of $\boldsymbol{\Lambda}_{ab}(\mathbf{x}, \mathbf{x}') \in \mathbb{R}^{N \times N}$, and refer to [3] or [4] for the details of $\boldsymbol{\lambda}_a(\mathbf{x}) \in \mathbb{R}^N$ and $\boldsymbol{\Psi}(\mathbf{x}) \in \mathbb{R}^{d_{\boldsymbol{\theta}} \times d_{\mathbf{y}}}$.

Since larger value of the mutual information indicates more informative, the term $q(\mathbf{x})$ in the running cost is defined using a certain monotonically decreasing function $v(\mathbf{x}) \geq 0$ as:

$$q(\mathbf{x}) = v\Big(\mathrm{I}\big[\boldsymbol{\theta}, \mathbf{y}|\mathbf{x}\big]\Big).$$

C. Belief Update Based on State Sequence

The optimal action is planned based on the present belief $p_n(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$ as described before, and then the state sequence $\mathbf{x}_{0:T}^n$ and observation $\mathbf{y}_{0:T}^n$ are obtained by executing the action for the target object. Based on Bayes' rule

$$p_{n+1}(\boldsymbol{\theta}) = \frac{p\big(\mathbf{y}_{0:T}^n|\mathbf{x}_{0:T}^n, \boldsymbol{\theta}\big)p_n(\boldsymbol{\theta})}{p\big(\mathbf{y}_{0:T}^n|\mathbf{x}_{0:T}^n\big)}$$

and Gaussian approximation of the marginal distribution $p\big(\mathbf{y}_{0:T}^n|\mathbf{x}_{0:T}^n\big)$, the mean and the covariance are updated as follows:

$$\begin{aligned} \boldsymbol{\mu}_{n+1} &= \boldsymbol{\mu}_n + \mathcal{C}_n\mathcal{S}_n^{-1}(\mathcal{Y}_n - \mathcal{M}_n), \\ \boldsymbol{\Sigma}_{n+1} &= \boldsymbol{\Sigma}_n - \mathcal{C}_n\mathcal{S}_n^{-1}\mathcal{C}_n^{\mathrm{T}} \end{aligned}$$

where $\mathcal{C}_n \in \mathbb{R}^{d_{\boldsymbol{\theta}} \times (T+1)d_{\mathbf{y}}}$, $\mathcal{S}_n \in \mathbb{R}^{(T+1)d_{\mathbf{y}} \times (T+1)d_{\mathbf{y}}}$, $\mathcal{Y}_n \in \mathbb{R}^{(T+1)d_{\mathbf{y}}}$, and $\mathcal{M}_n \in \mathbb{R}^{(T+1)d_{\mathbf{y}}}$ are defined as follows:

$$\begin{aligned} \mathcal{C}_n &= \left[\begin{array}{ccc} \mathbf{C}(\mathbf{x}_0^n) & \cdots & \mathbf{C}(\mathbf{x}_T^n) \end{array}\right], \\ \mathcal{S}_n &= \left[\begin{array}{ccc} \boldsymbol{\Phi}(\mathbf{x}_0^n, \mathbf{x}_0^n) & \cdots & \boldsymbol{\Phi}(\mathbf{x}_0^n, \mathbf{x}_T^n) \\ \vdots & \ddots & \vdots \\ \boldsymbol{\Phi}(\mathbf{x}_T^n, \mathbf{x}_0^n) & \cdots & \boldsymbol{\Phi}(\mathbf{x}_T^n, \mathbf{x}_T^n) \end{array}\right], \\ \mathcal{Y}_n &= \left[\begin{array}{ccc} (\mathbf{y}_0^n)^{\mathrm{T}} & \cdots & (\mathbf{y}_T^n)^{\mathrm{T}} \end{array}\right]^{\mathrm{T}}, \\ \mathcal{M}_n &= \left[\begin{array}{ccc} \Big(\mathbf{m}\big(\mathbf{x}_0^n\big)\Big)^{\mathrm{T}} & \cdots & \Big(\mathbf{m}\big(\mathbf{x}_T^n\big)\Big)^{\mathrm{T}} \end{array}\right]^{\mathrm{T}} \end{aligned}$$

where $\mathcal{C}_n$ and $\mathcal{S}_n$ are the cross-covariance matrix between $\mathbf{y}_{0:T}^n$ and $\boldsymbol{\theta}$, and the covariance matrix of $\mathbf{y}_{0:T}^n$ respectively. This is an extension of the one-sample belief update law described in [3].

The object recognition is achieved by repeating the optimal action planning and belief updating as described previously. The terminate condition is set by threshold e.g. $\|\boldsymbol{\Sigma}_n\|_{\mathrm{F}}^2 < \epsilon$ with the suitable threshold ϵ, or after $n_{\max}$ times repeat. Finally, the recognition result is obtained as the object corresponding to the nearest object parameter $\boldsymbol{\theta}$ in the database.

IV. EXPERIMENTS

A. Experiment 1: With Physical Simulator

1) Simulation Settings: We verify the effectiveness of our proposed scheme using the one link robot arm model shown in Fig. 2. The joint range is limited to $-\pi/2 \leq q \leq \pi/2$, and its equation of motion are discretized in a Euler integration manner with the sampling time $\Delta t = 0.01$[s]. We assume that the 2 DoF pressure sensor is mounted on the tip of the arm in order to obtain the observation. The reaction force model f_1 with the spring K and the damper D for the object as shown in Fig. 2 is supposed for the horizontal axis, and the dynamic friction model f_2 with the coefficient of dynamic friction μ' is also assumed for the vertical axis as follows:

$$f_1 = -(K\xi_x + D\dot{\xi}_x),$$
$$f_2 = -\mathrm{sign}(\dot{\xi}_y)\mu' f_1$$

where $\boldsymbol{\xi} = [\xi_x, \xi_y]^{\mathrm{T}}$ is the tip position of the arm.

In this experiment, the object recognition problem is regarded as the damper coefficient estimation problem: the object parameter $\theta = D$ is estimated using the exploratory action.

Let us describe how to learn the GP observation model. The spring coefficient and the dynamic friction coefficient are fixed as $K = 1$ and $\mu' = 0.5$ respectively, and we prepare 3 target objects $D \in \{1, 3, 5\}$. The observation $\mathbf{y} = \left[f_1, f_2\right]^{\mathrm{T}}$ and the state $\mathbf{x} = \left[q, \dot{q}\right]^{\mathrm{T}}$ are defined, and the training data $\mathcal{D}$ is constructed as follows: the range of state is set to $-\pi/2 \leq q \leq \pi/2$ and $-15 \leq \dot{q} \leq 15$, and a 15×15 grid is arranged at equal intervals on the range. We obtain the observation $\mathbf{y}$ for each grid point. The total number of training data is $N = 675$. The object's belief update is executed $n_{\max} = 10$ times.

Next, let us explain the settings for the information maximization control. The initial state is fixed to $\mathbf{x}_0 = \left[-\pi/2, 0\right]^{\mathrm{T}}$, and the length of the exploratory action is set to $T = 100$. The initial input sequence is set to $u_t = 3$ for $t = 0, 1, \ldots, T-1$. The running cost based on the belief $p_n(\theta)$ is defined as

$$\begin{aligned}\ell_n(t, \mathbf{x}, u) = &\exp\Big(-\rho^{(n)}\mathrm{I}\big[\theta, \mathbf{y}|\mathbf{x}\big]\Big) + ru^2 \\ &+ 1000\exp(-10(15 - \dot{q})) \\ &+ 1000\exp(-10(15 + \dot{q}))\end{aligned}$$

where $\rho^{(n)} > 0$ is a constant in order to change the maximization problem of the mutual information to the minimization problem, and experimentally $\rho^{(n)} = 10\exp(-0.2n^2)$ is used since the magnitude of mutual information will decrease by belief's updates. The 3rd and 4th terms in the running cost play a role of the penalty term as the angle velocity does not exceed the range of training data. The terminate cost is set to $h(\mathbf{x}) = \ell_n(T, \mathbf{x}, 0)$. The initial object's belief is given $p_0(\theta) = \mathcal{N}(3, 5^2)$, where its mean is equal to the mean of the object parameter in the training data.

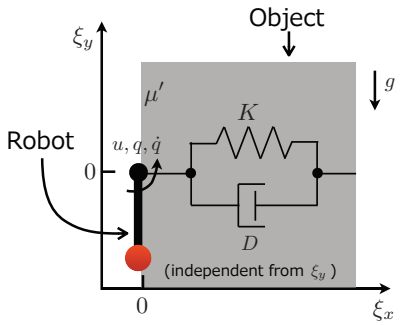


Fig. 2. Problem setting for Experiment 1. A tactile sensor is mounted on the tip of 1 DoF robot arm. As the object model, the spring-damper model is assumed for the horizontal direction, and the dynamic friction model is also assumed for the vertical direction.

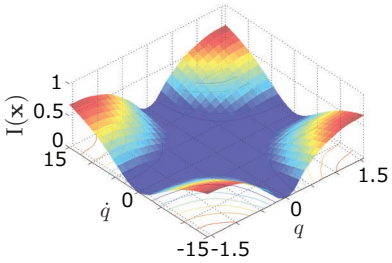
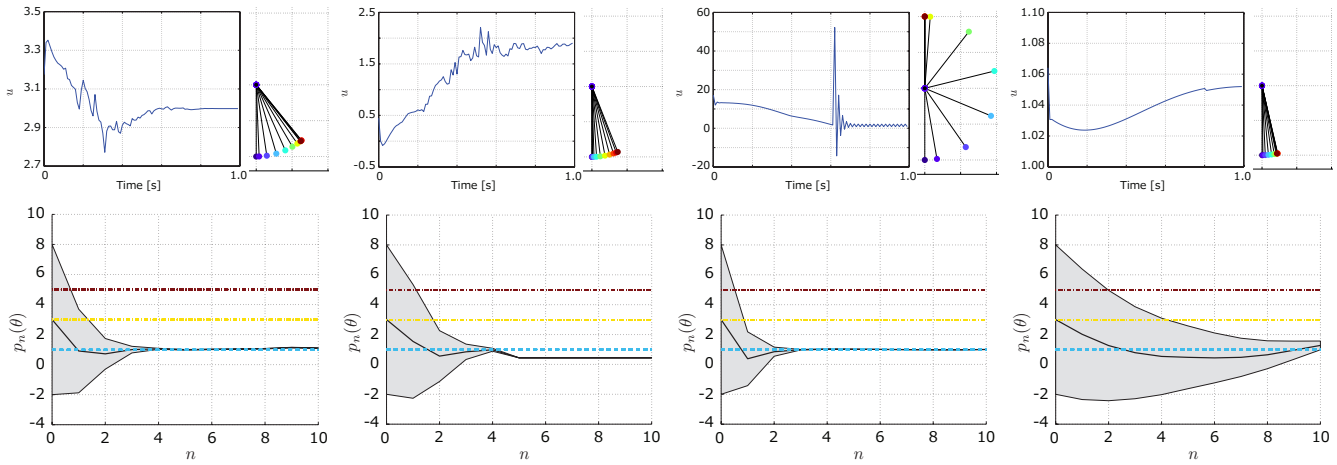


Fig. 3. Distribution of the mutual information based on the initial object's belief $p_0(\theta) = \mathcal{N}(3, 5^2)$.

2) Result: Firstly, the mutual information distribution based on the initial object's belief $p_0(\theta)$ computed using the GP observation model is shown in Fig. 3. The force from the damper depends on the velocity in the direction of the horizontal axis $\dot{\xi}_x$. The velocity $\dot{\xi}_x$ is gotten zero when the angle $q = 0$ and larger force from the spring is observed. Accordingly, the information about the damper could be buried in other information. Whereas, more information could be obtained when the absolute value of the angle $|q|$ is close to $\pi/2$. Consequently, we regard this distribution as appropriate.

Secondly, the recognition result is shown in Fig. 4. The input torque sequences obtained at $n = 0$ by our method are shown in the upper row of Figs. 4(a)-(b). For both cases with different values in r, the energy efficient and compliant exploratory actions are generated by the proposed method and the object recognition is successfully achieved. The balance in between the informativeness and energy efficiency is adjusted by the r: the larger value in r (Fig. 4(b)) generated energy-efficient actions, however, it is less informative as evidenced with the slower convergence of the belief updates than the smaller value (Fig. 4(a)). It was also confirmed that all the elements of local feedback gains $\mathbf{L}$ were relatively small for all the cases. Therefore, the generated controllers are compliant.

As the comparison, we implemented the PD controllers which generate the control input for achieving the desired state $\mathbf{x}_d = [\pi/2, 15]^{\mathrm{T}}$, and here this planning has done separately from control problem. The most informative action



(a) $r = 0.1$ (b) $r = 30$ (c) High Gain PD (d) Low Gain PD

Fig. 4. Results in Experiment 1. Regarding (a-b), more compliant but less informative actions are designed for the larger value in r. (c) This result indicates that the high gain PID controller gives informative actions. However, these are not compliant. (d) In contrast with (c), the actions are not informative but compliant. [Upper row] The left figure shows the input torque sequence of obtained exploratory action at $n = 0$, and the right figure shows the trajectories of arm for each action corresponds to the torques shown in the left figure. The color of the tip corresponds to the time. [Lower row] The horizontal axis shows the number of updates of the object's belief and the initial belief is set to $p_0(\theta) = \mathcal{N}(3, 5^2)$. The true object parameter is $D = 1$ shown using light blue line. The black line and gray region stand for the mean and standard deviation of the belief $p_n(\theta)$ respectively.

is obtained using the high gain PD controller as shown in Fig. 4(c) but the obtained action is energy inefficient since the large torque sequence is generated. In contrast, a more energy efficient action is obtained using the lower gain PD controller as shown in Fig. 4(d); nevertheless the convergence of the belief is slower as compared to the other methods since the planed action is infeasible by the controller.

These experimental results show that our proposed method can generate energy efficient and compliant exploratory behaviors.

B. Experiment 2: With Actual Robot

1) Experimental Settings: The effectiveness of our proposed scheme is validated by the robot hand system shown in Fig. 5(a). We prepared $L = 4$ objects as recognition targets shown in Fig. 5(b). This experiment was done with the robot hand (Shadow Dexterous Hand by Shadow Robot Company), and the tactile sensor mounted on its fingertip (BioTac by SynTouch) shown in Fig. 5(c). While this robot hand has 12 DoF, in this experiment we focused on 2 DoF, FFJ3 and FFJ4, as shown in Fig. 5(c) because of the limitation of the scalability (see Section V for the detailed discussion). These joints can generate actions that correspond to inflective and horizontal movements of the index finger, respectively. The angle position controller is provided with Robot Operating System (ROS), and the controller regulates the inner pressure of each pneumatic artificial muscles. Its maximum control rate and sensor data collection frequency are both 1000Hz. We describe below the details of the dynamics model and the observation model.

(a) Overview

(b) Target objects (c) Robot hand and tactile sensor

Fig. 5. Experimental settings for Experiment 2. (a) Overview of our robot hand system. The robot touches the objects on the turntable. (b) Target objects in this experiment. (c) 12 DoF robot hand and tactile sensor. Each joints are driven by pneumatic artificial muscles placed antagonistically. We use 2 DoF corresponding to inflective (pushing) and horizontal (rubbing) movements of the index finger respectively. The tactile sensor is mounted on the fingertip.

The dynamics model (2) of this robot hand is difficult to derive analytically because of the complex properties of the pneumatic artificial muscles. Instead, we identified it from training data using a nonlinear discrete-time ARX model whose nonlinearity is wavelet network and one-layer sigmoid network and whose sampling period is set to 0.01s. This are implemented in the MATLAB System Identification Toolbox. The state $\mathbf{x}$ here is $d_{\mathbf{x}} = 4$ dimensional $\mathbf{x} =$

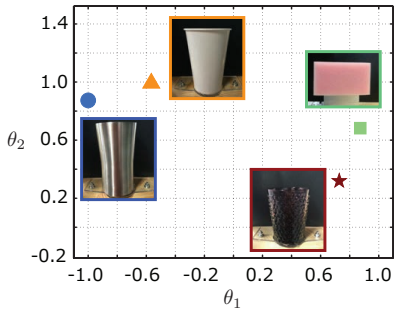


Fig. 6. Object parameters obtained by object manifold learning. In Experiment 2, the object parameters for all the objects are identified.

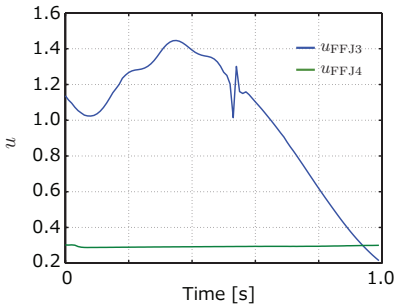


Fig. 7. Input of obtained action in Experiment 2. Positive values indicate that the joint moves a direction which increases the joint angle. The joint FFJ3 (blue) generates pushing movements, and the joint FFJ4 (green) generates rubbing movements. This input sequence lets the robot push and rub the object simultaneously.

$\left[q_{\mathrm{FFJ3}}, \dot{q}_{\mathrm{FFJ3}}, q_{\mathrm{FFJ4}}, \dot{q}_{\mathrm{FFJ4}}\right]^{\mathrm{T}}$, where q_{FFJ3} and q_{FFJ4} are the joint angle of FFJ3 and FFJ4 respectively, and $\dot{q}$ stands for each joint velocity. The joint angle ranges of FFJ3 and FFJ4 are $0 \leq q_{\mathrm{FFJ3}} \leq \pi/2$ and $-\pi/9 \leq q_{\mathrm{FFJ4}} \leq \pi/9$ respectively. Here, each input $u_j, j \in \{\mathrm{FFJ3}, \mathrm{FFJ4}\}$ is defined as the difference between the desired angle q_j^d and the actual angle q_j, $u_j = q_j^d - q_j$. In the training data collection, the desired angle was set to their maximum and minimum joint angles alternatively. The independency between joints are assumed, therefore, two dynamics models are separately learned for those joints. The total number of training data is 26,765.

The BioTac sensor gives pressure, vibration, and temperature as tactile information. In this experiment, $d_{\mathbf{y}} = 3$ dimensional tactile feature was used; 1-dimensional pressure data and 2-dimensional impedance data, all of which were obtained by using ROS. We collected 100-sample training data for each objects and the number of whole training data was 400. This data collection was done as follows: We design a random trajectories for each joint, and tens of thousands data is collected. And then, 100 data for each object is selected randomly. The object parameters were given by the object manifold learning scheme [4] as shown in Fig. 6.

Here, the initial state was fixed to $\mathbf{x}_0 = \left[\pi/12, 0, 0, 0\right]^{\mathrm{T}}$, and the length of the exploratory action was set to $T = 100$. The initial input sequence is set to $u_t = [0.2, 0.3]^{\mathrm{T}}$ for $t = 0, 1, \ldots, T-1$. The running cost was set to

$$\ell(t, \boldsymbol{x}, \boldsymbol{u}) = 10 \exp\Big(-\rho \mathrm{I}\big[\theta, \mathbf{y}|\mathbf{x}\big]\Big) + \mathbf{u}_t^{\mathrm{T}} \mathbf{R} \mathbf{u}_t + \sum_{j=1}^{4} \Big(\exp(-10(x_{j,\max} - x_j)) + \exp(-10(-x_{j,\min} - x_j))\Big)$$

where $\rho = 30$, $\mathbf{R} = 0.1\mathbf{I}$, $x_{j,\max}$ and $x_{j,\min}$ stand for the maximum and minimum values of the j-th entry of the state in the training data for GP model construction respectively.

The object's belief $p_0(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ was given by $\boldsymbol{\mu}_0 = [0.8, 0.5]^{\mathrm{T}}$ and $\boldsymbol{\Sigma}_0 = 0.2^2\mathbf{I}$, which means that it is uncertain whether the target object is O_2 (Dumpy Cup) or O_3 (Sponge). The exploratory action is designed under the conditions.

2) Results: The computed input sequence $\mathbf{u}$ is shown in Fig. 7. Here, the joint moves to a direction which increases its joint angle if the positive values are inputed, because u_j stands for $u_j = q_j^d - q_j$. As you can see in Fig. 7, the robot starts to push and rub simultaneously. Intuitively speaking, to discriminate dumpy cup and sponge, the finger should push the object to confirm its stiffness, and also rub the object to check the dumpiness. The obtained action is shown in Fig. 8. The upper row of Fig. 8 shows the action sequence at $t = 0, 20, \ldots, 100$ and the lower row shows the pose difference from the pose at $t = 0$. Inflective movements were firstly observed ($t = 0$ to $t = 40$) due to hardware properties and the initial state; the joint's movement gets slower if its angle is close to its limit, and the margin between the initial state and the angle limit of FFJ3 is wider than FFJ4. And then horizontal movements are observed ($t = 40$ to $t = 100$). It was also confirmed that the generated controllers are compliant since all the elements of local feedback gains $\mathbf{L}$ were relatively small. This movement can be interpreted as a suitable exploratory action to reduce the uncertainty in between the objects O_2 and O_3. The further investigation is required, but these results suggest that the effectiveness of our proposed method for exploratory action design in real environment.

V. DISCUSSIONS

We tackled the exploratory action design problem, and proposed information maximization control based on an optimal control approach. Our contribution is that the exploratory action design considering both the informativeness and the compliance is formulated as an optimal control problem. The effectiveness of our proposed method was investigated through experiments using simulated and real robots.

In some previous studies, similar methods have attempted to solve different problems as an optimal control problem. The active sensing problem (e.g. [11], [12]), such as field modeling of the environment is addressed with a mutual information criterion. However, contactless sensor, such as a laser rangefinder, or a vision sensor are targeted in those studies, in other words, compliance is not considered in the exploratory action design.

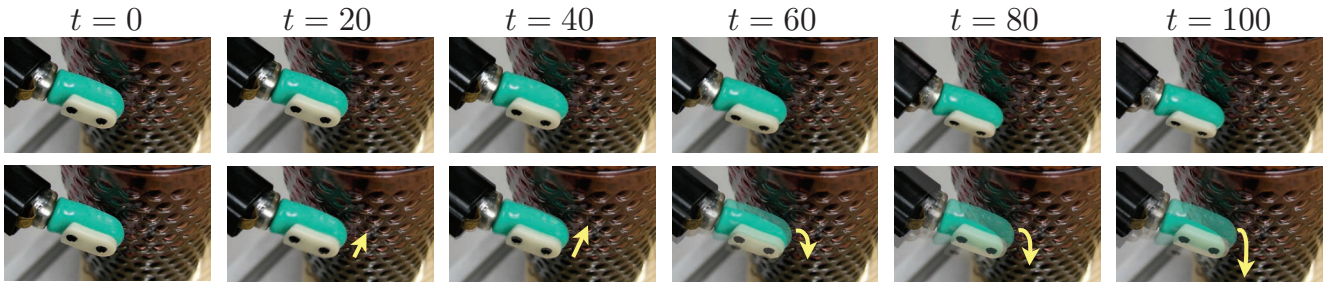


Fig. 8. Obtained action in Experiment 2. Intuitively speaking, to discriminate dumpy cup and sponge, the finger should push the object to confirm its stiffness, and also rub the object to check the dumpiness. The upper row shows the action sequence at $t = 0, 20, \ldots, 100$ and the lower row shows the pose difference from the pose at $t = 0$. The robot starts to push and rub simultaneously. Inflective movements are firstly observed ($t = 0$ to $t = 40$) due of hardware properties and the initial state, and then horizontal movements are observed ($t = 40$ to $t = 100$).

One of our future works will be an extension of scalability of the method. The computational complexity of our method is relatively large because of GP observation models. A simple method for the complexity reduction is to utilize approximation methods (e.g. [13], [14]). Using these simplifying method, we will validate the recognition rate of our proposed method. The effectiveness evaluation in more realistic scenarios would be interesting. We validated our method using 4 actual objects in the experiment. We are now conducting the experiments with more objects to investigate the scalability of our method. Furthermore, multimodal sensors are generally available on humanoid robots, so the extension of the method for such sensors will be also addressed.

APPENDIX

A. Definition of $\mathbf{\Lambda}$

Using the training data and hyperparamters, $\Lambda_{ab,ij}$ which is the (i,j)-th entry of $\mathbf{\Lambda}_{ab}$ is represented as follows:

$$\begin{aligned}\Lambda_{ab,ij} = \alpha_a^2 \alpha_b^2 \det\!\left(\left(\left(\mathbf{H}_a^{\boldsymbol{\theta}}\right)^{-1} + \left(\mathbf{H}_b^{\boldsymbol{\theta}}\right)^{-1}\right)\mathbf{\Sigma} + \mathbf{I}\right)^{-\frac{1}{2}} \\ \times \exp\!\left(-\tfrac{1}{2}\left(\boldsymbol{\theta}^{(i)} - \boldsymbol{\theta}^{(j)}\right)^{\mathrm{T}}\left(\mathbf{H}_{ab}^{\boldsymbol{\theta}}\right)^{-1}\left(\boldsymbol{\theta}^{(i)} - \boldsymbol{\theta}^{(j)}\right)\right) \\ \times \exp\!\left(-\tfrac{1}{2}\left(\boldsymbol{\theta}_{ab}^{ij} - \boldsymbol{\mu}\right)^{\mathrm{T}}\mathbf{R}_{ab}^{-1}\left(\boldsymbol{\theta}_{ab}^{ij} - \boldsymbol{\mu}\right)\right) \\ \times \exp\!\left(-\tfrac{1}{2}\left(\mathbf{x} - \mathbf{x}^{(i)}\right)^{\mathrm{T}}\left(\mathbf{H}_a^{\mathbf{x}}\right)^{-1}\left(\mathbf{x} - \mathbf{x}^{(i)}\right)\right) \\ \times \exp\!\left(-\tfrac{1}{2}\left(\mathbf{x}' - \mathbf{x}^{(j)}\right)^{\mathrm{T}}\left(\mathbf{H}_b^{\mathbf{x}}\right)^{-1}\left(\mathbf{x}' - \mathbf{x}^{(j)}\right)\right),\end{aligned}$$

where $\mathbf{H}_{ab}^{\boldsymbol{\theta}} = \mathbf{H}_a^{\boldsymbol{\theta}} + \mathbf{H}_b^{\boldsymbol{\theta}}$ and

$$\begin{aligned}\boldsymbol{\theta}_{ab}^{ij} &= \mathbf{H}_b^{\boldsymbol{\theta}}\left(\mathbf{H}_{ab}^{\boldsymbol{\theta}}\right)^{-1}\boldsymbol{\theta}^{(i)} + \mathbf{H}_a^{\boldsymbol{\theta}}\left(\mathbf{H}_{ab}^{\boldsymbol{\theta}}\right)^{-1}\boldsymbol{\theta}^{(j)}, \\ \mathbf{R}_{ab} &= \left(\left(\mathbf{H}_a^{\boldsymbol{\theta}}\right)^{-1} + \left(\mathbf{H}_b^{\boldsymbol{\theta}}\right)^{-1}\right)^{-1} + \mathbf{\Sigma},\end{aligned}$$

are defined.

REFERENCES

[1] J. A. Fishel and G. E. Loeb, "Bayesian exploration for intelligent identification of textures," *Frontiers in Neurorobotics*, vol. 6, no. 4, 2012.

[2] J. Sinapov, C. Schenck, K. Staley, V. Sukhoy, and A. Stoytchev, "Grounding semantic categories in behavioral interactions: Experiments with 100 objects," *Robotics and Autonomous Systems*, vol. 62, no. 5, pp. 632–645, 2014.

[3] H. Saal, J.-A. Ting, and S. Vijayakumar, "Active sequential learning with tactile feedback," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, 2010, pp. 677–684.

[4] D. Tanaka, T. Matsubara, K. Ichien, and K. Sugimoto, "Object manifold learning with action features for active tactile object recognition," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 608–614.

[5] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[6] D. J. C. MacKay, *Information Theory, Inference & Learning Algorithms*. New York, NY, USA: Cambridge University Press, 2002.

[7] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Athena Scientific, 2000.

[8] W. Li and E. Todorov, "Iterative linear quadratic regulator design for nonlinear biological movement systems." in *Proceedings of International Conference on Informatics in Control, Automation and Robotics*, 2004, pp. 222–229.

[9] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. MIT Press, 2006.

[10] M. P. Deisenroth, M. F. Huber, and U. D. Hanebeck, "Analytic moment-based Gaussian process filtering," in *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, pp. 225–232.

[11] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky., and H. F. Durrant-Whyte, "Information based adaptive robotic exploration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002, pp. 540–545.

[12] J. Le Ny and G. J. Pappas, "On trajectory optimization for active sensing in Gaussian process models," in *Proceedings of the Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, 2009, pp. 6286–6292.

[13] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," in *Advances in Neural Information Processing Systems 18*, 2006, pp. 1257–1264.

[14] Y. Shen, M. Seeger, and A. Y. Ng, "Fast Gaussian process regression using KD-Trees," in *Advances in Neural Information Processing Systems 18*, 2006, pp. 1225–1232.