SofRobotics

Design of physical user-robot interactions for model identification of soft actuators on exoskeleton robots



The International Journal of Robotics Research 1–14 © The Author(s) 2019 © © © © Article reuse guidelines: sagepub.com/journals-permissions DOI: 10.1177/0278364919853618

journals.sagepub.com/home/ijr



Masashi Hamaya^{1,2}, Takamitsu Matsubara^{1,3}, Tatsuya Teramae¹, Tomoyuki Noda¹ and Jun Morimoto¹

Abstract

Recent breakthroughs in wearable robots, such as exoskeleton robots with soft actuators and soft exosuits, have enabled the use of safe and comfortable movement assistance. However, modeling and identification methods for soft actuators used in wearable robots have yet to be sufficiently explored. In this study, we propose a novel approach for obtaining accurate soft actuator models through the design of physical user–robot interactions for wearable robots, in which the user applies external forces to the robot. To obtain an accurate soft actuator model from the limited amount of data acquired through an interaction, we leverage an active learning framework based on Gaussian process regression. We conducted experiments using a two-degree-of-freedom upper-limb exoskeleton robot with four pneumatic artificial muscles (PAMs). Experimental results showed that physical interactions between the exoskeleton robot and the user were successfully designed to allow PAM models to be identified. Furthermore, we found that data acquired through an interaction could result in more accurate soft actuator models for the exoskeleton robots than data acquired without a physical interaction between the exoskeleton robot and the user.

Keywords

Soft actuators, exoskeleton robots, physical human-robot interactions, human-in-the-loop, Gaussian process, active learning

1. Introduction

Wearable robots can be used in various situations, including assistance with daily life or with rehabilitation (Dollar and Herr, 2008; Gopura et al., 2016; Yan et al., 2015). Recent studies have developed exoskeleton robots with soft actuators and soft exosuits. These soft actuators, which include series elastic actuators (SEAs) (Wang et al., 2015; Yu et al., 2015), variable stiffness actuators (VSAs) (Mghames et al., 2017) or pneumatic artificial muscles (PAMs) (Cao et al., 2018; Ferris et al., 2005; Liu et al., 2017), have attracted significant attention owing to their safe and compliant physical properties suitable for wearable robots. The soft exosuits (Asbeck et al., 2015; Ding et al., 2018) are composed of soft and lightweight textiles, and can thus easily fit to the shape of the user's body.

However, modeling and identification methods for soft actuators used in wearable robots have yet to be sufficiently explored. In these actuators or suits composed of soft materials, force transmission mechanisms tend to introduce highly nonlinear dynamics (Asbeck et al., 2015; Lagoda et al., 2010; Liu et al., 2017; Mghames et al., 2017). Moreover, they are stretched by external forces because of the compliance. Therefore, rich datasets generated using various external forces are required to identify such models that include nonlinearity and compliance. The standard approach is to apply a load to an actuator by placing various weights (Merola et al., 2018), or to employ antagonistic muscles (Teramae et al., 2013) to generate external forces. However, such approaches do not necessarily take into account actual situations in which the robot is worn by a human user.

In our approach, the external forces applied to soft actuators are generated through user-robot physical interactions

Corresponding author:

¹The Department of Brain Robot Interface, ATR-CNS, Kyoto, Japan
²The Graduate School of Frontier Bioscience, Osaka University, Osaka, Japan

³The Graduate School of Science and Technology, Nara Institute of Science and Technology, Nara, Japan

Masashi Hamaya, The Department of Brain Robot Interface, ATR-CNS, 2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan. Email: hamaya@atr.jp



Fig. 1. Schematic diagram of our proposed method. We designed user–robot interactions by providing the human user with a way to generate external force based on the current status (e.g., model uncertainty) of the learned model. Our proposed system designs action patterns for the human user and robot based on the current soft actuator model, and then asks the user to execute actions by physically moving the wearable robot. Because action patterns for the human user and robot actuators equipped in the wearable robot. Data generated through physical user–robot interaction are used to update the soft actuator model.

to tackle this model identification problem. In other words, we developed a user-robot interaction design strategy to efficiently generate external forces for identifying soft actuator models from a limited number of data collection trials. Accordingly, we collected data under versatile circumstances close to those in practical use for the model identification, and avoided collecting unnecessary data through unnecessary experimental trials.

Figure 1 shows a schematic diagram of our proposed method. We designed the user–robot interactions by providing a human user with a way to generate an external force based on the current status (e.g., model uncertainty) of the learned model. Our proposed system designs action patterns for the human user and robot based on the current soft actuator model, and then asks the user to execute the actions by physically moving the wearable robot. Because the action patterns for the human user and robot are different, the user ends up applying external forces to the soft actuators equipped in the wearable robot. The data generated through physical user–robot interaction are used to update the soft actuator model.

In this study, we used PAMs as the soft actuators. To design the user–robot interaction and acquire a soft actuator model, we leveraged a Gaussian process (GP) regression and GP active learning framework. GP can express the model uncertainty as the predictive variance and nonlinearity of a PAM. GP active learning sequentially selects a data point, which reduces the model uncertainty. It can be used to accelerate the PAM model identification as well as avoid overfitting. In our learning framework, the reference trajectories that maximize the information gain are provided to the human user and robot. The user then tries to follow the provided trajectory to generate external forces used to update the GP-based PAM (GP-PAM) model efficiently.

In our preliminary study (Hamaya et al., 2017b), we explored how the GP active learning framework can be useful for soft actuator identification using a simple onedegree-of-freedom (one-DoF) elbow-joint assistive robot (Noda et al., 2013). Although the GP active learning framework was successfully applied, the applicability of our previous method was limited to identifying only one PAM model attached to the one-DoF assistive robot, owing to the necessity of manually tuning GP the hyperparameters that determine the shape of the approximated function.

In this study, we newly propose a GP active learning framework that can be applicable to further practical situations in which a multi-DoF robot and multiple actuators are involved. The contributions of this study are as follows.

- 1. We developed a systematic procedure to simultaneously identify multiple soft actuator models on a wearable robot through a physical human-robot interaction.
- 2. We newly adopted an active learning method that can learn the hyperparameters of the GP-PAM models such that the shape of the approximated function can be adaptively learned through the learning process.
- We conducted model identification experiments using a 2-DoF upper-limb exoskeleton robot with four PAMs. The results demonstrate that our method outperforms standard model identification approaches that do not take the user-robot interactions into account.

Our proposed method is composed of two components: the human-in-the-loop optimization and GP active learning. However, they are not independent contributions; rather, its combination is essentially important because our humanin-the-loop data collection process requires the criterion of model uncertainty, and it requires the GP-PAM model owing its capability for evaluating the model uncertainty. The remainder of the paper is organized as follows. Section 2 describes related studies regarding physical userrobot interaction designs and PAM-driven exoskeleton robots. Section 3 introduces our proposed method. Section 4 describes our experimental setup. Section 5 provides the experimental results. Section 6 discusses an additional offline analysis. Finally, Section 7 provides some concluding remarks regarding this research.

2. Related work

In this section, we describe recent physical user-robot interaction and user-in-the-loop optimization methods for wearable robots. We also present PAM-driven exoskeleton robots and discuss the model identification methods developed.

2.1. Physical user-robot interaction design

Physical user-robot interactions have been investigated in collaborative manipulation tasks. Ghadirzadeh et al. (2016) proposed the modeling of physical user-robot collaborations, and the use of such modeling to select the control output through Bayesian optimization. Rozo et al. (2016) developed a task-parameterized interaction model and a stiffness estimation method for a transport task. Peternel et al. (2016b) utilized human user's electromyography (EMG) to modulate the joint stiffness of a robot for a collaborative cut-sawing task. Baicsy et al. (2017) proposed estimating the objective function of a manipulation task by observing how a user physically guides a robot to generate hand trajectories. Maeda et al. (2017) estimated the user's movement intentions and generate robot joint trajectories through the use of dynamic movement primitives in handover tasks. Khoramshahi and Billard (2018) employed taskparameterized dynamical systems and switched the dynamical systems according to the user's movement intentions. The above methods designed interactions for completing collaborative tasks. The robots optimize their behavioral strategies according to the users' movement intentions. In contrast, in our study, human users provide necessary data to identify a soft actuator model according to the uncertainty of the actuator model learned using the robots.

A way for designing physical user–robot interactions to enhance the user's motor performance has also been recently explored. User-in-the-loop frameworks have been proposed to optimize the assistive strategies through physical user–robot interactions. These user-profiling assistive strategies have successfully been used to reduce metabolic costs when walking (Ding et al., 2018; Koller et al., 2016; Zhang et al., 2017), reduce muscle activities (Hamaya et al., 2017a), or generate preferred walking movements on an active prosthesis (Thatte et al., 2017).

In this study, we utilized a user-in-the-loop framework for the actuator model identification, rather than for optimizing assistive strategies. The user-in-the-loop framework can be extremely useful in acquiring model identification data because the identified model needs to represent the actuator behaviors when a user is wearing a robot. However, difficulties are encountered when using the userin-the-loop framework for the model identification, because we need to ask human users to conduct many movement trials to provide a sufficient amount of data. In addition, it is not clear what data collection methods of user-robot physical interaction data are required to improve the actuator models.

To solve these problems, we propose an active-learning based approach using GP models (Garnett et al., 2014) to efficiently design a physical user–robot interaction and provide concrete instructions to the users regarding how to generate data for model identification such that the human users do not need to conduct too many movement trials.

2.2. PAM-driven exoskeleton robots

PAMs have attracted significant attention as an actuator for a wearable robot owing to their compliance and safety with a high power-to-weight ratio. Ferris et al. (2005) proposed controlling the PAMs in proportion to the amplitude of the user's EMG signals without conducting model identification. Liu et al. (2017) used a static PAM model under the assumption that the muscle tension linearly changes with respect to the input pressure at the same muscle length. The parameters of the model were identified through modulation of the muscle length, contraction force, and pressure. The antagonistic muscle was used to provide external forces for the identification process.

Cao et al. (2018); Merola et al. (2018); Park et al. (2014) used linear dynamical models. Teramae et al. (2018) used a static PAM model proposed for position control and employed a PAM model identification method proposed by Teramae et al. (2013). Ugurlu et al. (2015) proposed the use of nonlinear dynamical PAM models and applied them to the control of a lower-limb exoskeleton robot Ugurlu et al. (2016). Koller et al. (2016), Peternel et al. (2016a), and Hamaya et al. (2017a) developed black-box optimization methods for the control of PAM-driven exoskeleton robots without conducting PAM model identification.

As introduced above, most of the previously proposed PAM-driven exoskeleton robot control methods have used parametric PAM models to determine the pressure command to generate the desired contraction force. Otherwise, PAM controllers are directly derived through learning trials to accomplish the given tasks without identifying the PAM model. These parametric PAM models, which are typically derived through a consideration of the physical property of the PAM mechanisms, cannot fully capture the behavior of the PAMs owing to the occurrence of modeling errors between the real PAMs and the method used to parameterize them. Therefore, in our study, we introduced a GP-based PAM model to represent the mechanical and dynamical properties of PAMs using data sampled from a real system. In contrast, we were able to apply direct PAM controller learning approaches for PAM-driven exoskeleton robot



Fig. 2. Overview of our implementation for physical user–robot interactions to identify soft actuator models: (1) we learn GP models of actuators; (2) select optimal interaction strategy that maximizes mutual information from pre-constructed user–robot trajectory bank by GP active learning; and (3) user and robot execute interactions to collect the soft actuator's data.

control. However, when we use this approach, the PAM controllers need to be learned for each target task. Therefore, this approach is not practical if we need to use an exoskeleton robot for a wide variety of different tasks. In our study, we identified a GP-PAM model through physical user–robot interactions such that the PAM-driven exoskeleton robot can generate the desired force according to the target problems.

3. Proposed framework

We introduce the interaction design strategy for the collaborative soft actuator model identification with the exoskeleton robot. The schematic diagram of our implementation is depicted in Figure 2: (1) identifying GP models of the soft actuators from collected data; (2) selecting an interaction strategy by the GP active learning algorithm from a pre-constructed user–robot trajectory bank, and (3) the user is asked to track the proposed reference joint trajectory and the robot generates joint torques using the selected pressure command to collect the data of the soft actuators. We repeat the procedure above to improve the soft actuator model.

3.1. GP soft actuator model identification

The soft actuator model is assumed as

$$y_t = f(\mathbf{x}_t) + \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \sigma_n^2)$$
 (1)

where x and y are inputs (e.g., actuator's length, velocity, input commands) and outputs (e.g., contraction force) of the soft actuator, and ϵ is an additive Gaussian noise. We

applied GP regression (Rasmussen and Williams, 2006) to model the soft actuator dynamics. The GP can represent the model uncertainty with the predictive variance; thus, we can derive the entropy of the acquired model distribution analytically. With a input $X = [x_1, \ldots, x_n]$ ($x \in \mathbb{R}^d$) and output $y = [y_1, \ldots, y_n]^\top$ ($y \in \mathbb{R}$) dataset, a predictive distribution of $f(x_*)$ for the new input x_* can be represented as follows:

$$p(f(\boldsymbol{x}_*)|\boldsymbol{X},\boldsymbol{y},\boldsymbol{\theta}) = \mathcal{N}\left(f(\boldsymbol{x}_*)|m_{f|\boldsymbol{\theta}}(\boldsymbol{x}_*),\sigma_{f|\boldsymbol{\theta}}^2(\boldsymbol{x}_*)\right)$$
(2)

where θ is a hyperparameter vector and the mean and the variance can be derived as (Murphy, 2013)

$$m_{f|\boldsymbol{\theta}}(\boldsymbol{x}_*) = \boldsymbol{k}_*^{\mathrm{T}} (\boldsymbol{K} + \sigma_n^2 \mathbf{I})^{-1} \boldsymbol{y}$$
(3)

$$\sigma_{f|\boldsymbol{\theta}}^{2}(\boldsymbol{x}_{*}) = k_{**} - \boldsymbol{k}_{*}^{\mathrm{T}}(\boldsymbol{K} + \sigma_{n}^{2}\mathbf{I})^{-1}\boldsymbol{k}_{*}$$

$$\tag{4}$$

Here, $k(\cdot)$ is a kernel function $(\mathbf{k}_* := k(\mathbf{X}, \mathbf{x}_*))$, $k_{**} := k(\mathbf{x}_*, \mathbf{x}_*))$, \mathbf{K} is a kernel matrix, each of whose element follows $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. In this study, we used a squared exponential kernel:

$$k(\boldsymbol{x}_p, \boldsymbol{x}_q) = \sigma_k^2 \exp\left(-\frac{1}{2}(\boldsymbol{x}_p - \boldsymbol{x}_q)^{\mathrm{T}} \boldsymbol{\Lambda}(\boldsymbol{x}_p - \boldsymbol{x}_q)\right) \quad (5)$$

where $\mathbf{\Lambda} = \text{diag}[a_1^{-1}, \dots, a_d^{-1}]$ is a precision matrix and controls the horizontal scale over which the function changes and σ_k controls the vertical scale of the function (Murphy, 2013). We define the hyperparameter vector as $\boldsymbol{\theta} = [a_1, \dots, a_d, \sigma_k, \sigma_n]^{\mathrm{T}}$.

3.2. User-robot interaction strategy

3.2.1. Defining reference trajectories as candidate data sampling points. Before we collect the user-robot interaction data, we define a set of reference trajectories as candidate data sampling points for the active learning framework. These reference trajectories for the human user and the robot must include the ranges where the actuators operate for the target tasks. In our previous work (Hamava et al., 2017b), we simply designed the reference trajectories based on the mechanically reachable area of a single-DoF robot. However, for multi-DoF robots, this simple reference design strategy would generate many unnecessary trajectories that include irrelevant movement ranges for the target tasks. Thus, in this study, we came up another approach: we prepared the reference trajectories by moving a human experimenter's arm while he was wearing an arm exoskeleton robot given different pressures. The resulting reference trajectories naturally represented the range of the human experimenter's movements that are close to practical situations.

Here, the set of reference trajectories is represented as $\mathcal{X}_{\text{ref}} = \left\{ \boldsymbol{X}_{\text{ref}}^{(1,1)}, \cdots, \boldsymbol{X}_{\text{ref}}^{(J,M)} \right\}, \text{ where each trajectory is defined as } \boldsymbol{X}_{\text{ref}}^{(j,m)} = \left[\boldsymbol{x}_{1}^{(j,m)}, \dots, \boldsymbol{x}_{t}^{(j,m)}, \dots, \boldsymbol{x}_{T}^{(j,m)} \right].$ $i = 1, \ldots, J$ is an index of a trajectory, and $m = 1, \ldots, M$ is an index of an actuator.

3.2.2. GP active learning for designing user-robot interactions. We applied a GP active learning method to select the reference trajectory for designing physical interactions. Kapoor et al. (2007) and Tanaka et al. (2014) demonstrated that efficient object categorizations could be achieved by GP active learning methods. However, most of the GP active learning methods have been implemented with fixed GP hyperparameters while these hyperparameters were manually designed based on the experts' knowledge. If these predetermined hyperparameters are inappropriate for the learning procedure, they can result in the overfitting of the model to the training dataset. Therefore, we leveraged an active learning of GP hyperparameters (Garnett et al., 2014). Based on uncertainty of the hyperparameters, the algorithm actively selects the reference trajectory to update the hyperparameters.

To select a new data point, the mutual information between the hyperparameter θ and mean output of the model f(x) was used as the objective function:

$$I(f(\mathbf{x});\boldsymbol{\theta}) = H(f(\mathbf{x})) - H(f(\mathbf{x})|\boldsymbol{\theta})$$
(6)

where $H(\cdot)$ is the entropy. We can derive the entropy $H(f(\mathbf{x})|\boldsymbol{\theta})$ because $p(f(\mathbf{x})|\mathbf{X}, \mathbf{y}, \boldsymbol{\theta})$ is a Gaussian distribution as in (2). Meanwhile, the entropy H(f(x)) cannot be easily obtained because the marginal distribution $p(f(\mathbf{x})|\mathbf{X}, \mathbf{y}) = \int p(f(\mathbf{x})|\mathbf{X}, \mathbf{y}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathbf{X}, \mathbf{y}) d\boldsymbol{\theta}$ cannot be analytically derived. We then calculate the approximated marginal distribution q(f(x)|X, y) from integrating an

approximated linear Gaussian model of $p(f(x)|X, v, \theta)$ with respect to $\boldsymbol{\theta}$ and an approximated posterior of the hyperparameter $q(\boldsymbol{\theta}|\boldsymbol{X},\boldsymbol{y})$. Finally, we can obtain the approximated form of (6) as follows Garnett et al. (2014):

$$I(\mathbf{x}) \simeq \ln \frac{\sigma_f^2(\mathbf{x})}{\sigma_{f|\theta}^2(\mathbf{x})},\tag{7}$$

where $I(f(x); \theta)$ is denoted as I(x) for simplicity. Here $\sigma_{f|a}^2(x)$ is the predictive variance given the hyperparameters $\boldsymbol{\theta}$. We utilized $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$ to evaluate the predictive variance, where $\hat{\theta}$ was estimated by the *maximum a posteriori* as in Appendix A. $\sigma_f^2(x)$ is the predictive variance of the marginalized distribution and can be derived as presented in Appendix B. We applied the nearest Semi Positive Definite (SPD) algorithm that approximates the covariance matrix to the nearest positive define matrix to avoid numerical problems pertaining to optimization of the hyperparameter $\boldsymbol{\theta}$. The algorithm subsequently selects reference trajectories to conduct user-robot interactions as follows:

$$L_{\mathrm{I}}\left(\boldsymbol{X}_{\mathrm{ref}}^{(j,\cdot)}\right) = \sum_{m=1}^{M} \sum_{t=1}^{T} I\left(\boldsymbol{x}_{t}^{(j,m)}\right)$$
(8)

$$j^* = \arg\max_{j} L_{\mathrm{I}}\left(\boldsymbol{X}_{\mathrm{ref}}^{(j,\,\cdot)}\right) \tag{9}$$

We summarize the algorithm of our proposed method in Algorithm 1. First, we prepare the priors of the hyperparameters for M actuators, initial dataset, and reference trajectories. The hyperparameters of the GP models were estimated by the maximum a posteriori with using the dataset and priors $p(\theta)$ as described in Appendix A (line 2). We compute the mutual information Garnett et al. (2014) between the function f(x) and the hyperparameters θ at each sample point for each actuator (line 3). Then, the index of reference trajectory j^* is selected to design userrobot interaction that maximizes the mutual information as in (9) (line 4). Subsequently, the user and robot generate movements by following the reference trajectories. Then, input X_s and output v_s data are sampled (line 5). The newly sampled data is merged to the input X and output v dataset (line 6). When we merge the newly acquired data, we

Algorithm 1 User-robot collaborative model identification

Require: Input X and output y dataset; Hyperprior $p(\theta)$; Reference trajectories \mathcal{X}_{ref}

- 1: for physical interaction trials do
- $\boldsymbol{\theta} \leftarrow \text{LearnGPmodels}(p(\boldsymbol{\theta}), \boldsymbol{X}, \boldsymbol{y})$ 2:
- $I(\mathbf{x}) \leftarrow \text{MutualInformation}(p(f(\mathbf{x})|\mathbf{X}, \mathbf{y}, \boldsymbol{\theta}), p(\boldsymbol{\theta}|\mathbf{X}, \mathbf{y}), \hat{\boldsymbol{\theta}})$ 3:
- $j^* \leftarrow \text{SelectInteraction}(I(\mathbf{x}), \mathcal{X}_{ref}) \\ X_s, y_s \leftarrow \text{DataSampling}(X_{ref}^{(*, \cdot)})$ $4 \cdot$
- 5:
- Add the pair of newly sampled input X_s and output y_s data 6: to the dataset X and y after a data subsampling procedure.
- 7: end for
- 8: return $\hat{\theta}$



Fig. 3. Experimental setup. (a) Experimental platform composed of exoskeleton robot and visual feedback system. (b) Upper-limb exoskeleton robot. Our exoskeleton robot contains four DoFs: shoulder abduction/adduction (SAA), shoulder flexion/extension (SFE), elbow flexion/extension (EFE), and wrist flexion/extension (WFE). In this study, we utilized two DoFs, SFE and EFE, out of four. To move these two joints, we activated four PAMs.

applied a data subsampling procedure as explained in Appendix C.

4. Experiments

To evaluate our proposed method, we conducted PAM model identification experiments and an offline analysis. We utilized two joints out of our four-DoF upper-limb exoskeleton robot driven by PAMs. Two healthy adults participated in the experiments. All participants were provided written informed consent before participation. Wearing the exoskeleton robot, the participants were asked to track the reference trajectories selected by the active GP algorithm while the robot also used the selected desired pressure profiles to collect data for the identification (Figure 3(a)). For comparison, the standard PAM identification procedure that does not consider the user–robot interaction was employed. We further conducted an offline analysis to investigate how the tracking errors affected the model identification performances.

4.1. Experimental setup

We used our upper-limb exoskeleton robot (Furukawa et al., 2017; Noda et al., 2014), as shown in Figure 3(b). Our exoskeleton robot contains four DoFs: shoulder abduction/adduction (SAA), shoulder flexion/extension (SFE), elbow flexion/extension (EFE), and wrist flexion/extension (WFE). In this study, we utilized two DoFs, SFE and EFE, out of the four joints. To move these two joints, we activated four PAMs, and two sets of agonist and antagonist PAMs. The agonist muscles used 40 mm PAMs (Festo MAS-40) and the antagonist muscles used 10 mm PAMs (Festo MAS-10). The servo valves are equipped with each muscle. The contraction force of each PAM is transmitted to a robot joint via Bowden cables. Encoders are mounted on each joint to measure the joint angles. Load cells are attached to the tips of the PAMs to measure the contraction

force. The PAM states were defined as $\mathbf{x} = [L, \dot{L}, P]^{\top}$, where *L* is the muscle length and \dot{L} is the change rate of the length. Here *P* is the pressure command to the servo valve. The output of the PAM model *y* is the contraction force.

We provided visual feedback to the participants. The reference joint angles and the current posture of the robot were displayed using the Open GL library. The red link in Figure 3(a) shows the reference joint angles and the green one shows the current robot posture.

4.2. Defining a set of reference trajectories

In this experiment, we defined the reference trajectories by moving a human experimenter's arm while he was wearing the arm exoskeleton robot. Simultaneously, the robot was activated by generating the pressure command to the PAMs. The rth reference pressure profile $\mathbf{P}_r = [P_r(0), \ldots, P_r(t), \ldots, P_r(T)]$ was represented with a sinusoidal function $P_r(t) = A \sin(2\pi\alpha t) + B$, where the duration for one movement trial T was 4.0 s. We designed sinusoidal the parameters of the function as $A = [-5/3, 5/3], B = [8/3, 13/3], \text{ and } \alpha = 0 \ (t < 1.0),$ $0.5 (1.0 \le t < 2.0), 0.25 (2.0 \le t \le 4.0)$ Hz such that these parameter ranges encompass the region where the robot operates. This yields four sinusoidal input pressure patterns for each PAM. Because we used four PAMs, we had to prepare 256 ($=4^4$) pressure command profiles. The sampling frequency was 250 Hz. In a later section, we show the actual profile of the reference joint angle trajectories and the pressure command profiles in Figure 5.

When we collected the data for the PAM model identification, we also measured the force applied to the PAMs. However, the force data were only used for evaluating the PAM model learning process in the offline analysis. In other words, activating the PAMs is not necessary for preparing the reference joint trajectories and pressure command profiles.



Time

Fig. 4. Snapshots of the user–robot interactive PAM model identification experiment. Wearing an exoskeleton robot, participants were asked to track reference trajectories selected by an active GP algorithm while the robot used selected desired pressure profiles to collect data for identification. We provided visual feedback to the participants. Reference joint angles and the current posture of the robot were displayed using Open GL library. Red links show reference joint angles and green links show the current robot posture.



Fig. 5. Example sequence of selected reference trajectories of shoulder joint and pressure command profiles of PAM1 by our active GP learning method for one experimental sequence. (Top) Selected reference trajectories. (Bottom) Pressure command profiles.

4.3. Designing user–robot interaction for PAM model identification

In this study, we adopted a Matlab software package for the GP hyperparameter active learning (Garnett et al., 2014).² The duration for one trial was 4.0 s. The participants conducted 10 trials to generate the physical user–robot interaction data.

For comparison, we employed a standard PAM model identification procedure that does not consider the userrobot interaction. We adopted a linear parametric PAM model and utilized an automatic PAM identification technique (Teramae et al., 2013) in which antagonist PAMs were used for generating the external forces. The details of the model and the identification process are presented in Appendix D. We evaluated the force output estimation error of the learned PAM models on a test dataset. As the test dataset, we utilized all of the data points that were collected when we prepared the reference trajectories, as described in Section 4.2.

5. Results

5.1. PAM model identification through user–robot interactions

The participants observed the current desired posture displayed on the monitor that follows the reference trajectories selected by the GP active learning method (Figure 4). Simultaneously, the participants were asked to track the displayed posture by moving their arm with the exoskeleton robot through a physical user–robot interaction. Figure 5 shows an example of a sequence of selected reference trajectories of the shoulder joint, and the pressure command



Fig. 6. Learning curves of our proposed approach and that of the standard PAM procedure. The vertical axis shows the force prediction error rate. Error rates were derived by normalizing the error with the maximum contraction force of test data. Error bars show standard deviations of predicted errors on eight experiments. Red lines show learning performances of our proposed method. Gray lines show that of the standard identification method, which does not consider user–robot interactions. Our proposed method successfully learned the PAM model through physical user–robot interactions. Although error rates were reduced even with the standard identification process, progresses of model learning were significantly worse than with our proposed approach. This result indicates that user–robot interaction successfully enhanced progress of actuator model identification.

profiles of the PAM1 by our active GP learning method for one experimental sequence.

Figure 6 shows the learning curves of our proposed approach and that of the standard PAM procedure. The total number of the data points to the represent GP-PAM model was 30. The vertical axis shows the force prediction error rate. The error rates were derived by normalizing the error with the maximum contraction force of the test data. The error bars show the standard deviations of the predicted errors on eight experiments. The red lines show the learning performances of our proposed method. The gray lines show that of the standard identification method that does not consider the user-robot interaction. Our proposed method successfully learned the PAM model through the physical user-robot interactions. Although the error rates were reduced even with the standard identification process, The progresses of model learning were significantly worse than with our proposed approach. This result indicates that user-robot interactions successfully enhanced the progress of actuator model identification.

5.2. Comparisons of actuator model identification performances

Figure 7 shows the comparisons of the actuator model identification performances. The red bar shows the contraction force prediction performance of our proposed method with the two participants. The dark gray bar shows that of the standard linear PAM model without user-robot



Fig. 7. Comparisons of actuator model identification performances. The red bar shows contraction force prediction performance of our proposed method with two participants. The dark gray bar shows that of the standard linear PAM model without user-robot interactions. The light gray bar shows that of the GP-PAM model without user-robot interactions. An ordinal GP-PAM model with non-interactive batch training cannot be applied for whole training data due to its heavy computational burden since it does not include any subsampling methods unlike our proposed method. To cope with this problem, we adopted SPGP model for this comparison. Our proposed method outperformed two other standard non-interactive identification procedures. This result indicates importance of user-robot physical interactions to acquire better actuator models for wearable robots.



Fig. 8. Model identification performances with different participants. Error bar shows standard deviation. We applied Welch's *t*-test and did not find any statistically significant difference between the prediction performances of the models acquired through interactions with two different participants. This result indicates that the performances are comparable between these two users in the experiment.

interactions. The light gray bar shows that of the GP–PAM model without user–robot interactions. Because ordinal GP model non-interactive batch learning with many data points cannot be used because of heavy computational burden, we adopted the sparse GP (SPGP) model (Snelson and Ghahramani, 2006) for this comparison. A detailed explanation is provided in Appendix E. Our proposed method outperformed two other standard non-interactive identification procedures. This result indicates the importance of physical user–robot interactions to acquire better actuator models for wearable robots.

In addition, to validate how our proposed actuator model identification procedure with physical user–robot interactions can be used to operate properly on different participants, we individually evaluated the contraction force prediction performances of the acquired model through the interaction of the two different participants. We used the same set of reference trajectories and the pressure command profiles for the experiments with these two participants.

Figure 8 shows the model identification performances with different participants for eight experiments. The error bar shows the standard deviation. We applied Welch's *t*-test (5% significance level) and did not find any statistically significant difference between the prediction performances of the models acquired through the interactions with the two different participants. This result indicates that the performances are comparable between these two users in the experiment.

5.3. Investigation of how tracking errors affect model identification performances

In our user-robot interaction experiment, the human participant was asked to track the reference trajectories that

were selected by our active GP learning method. However, precisely following the displayed reference movements is difficult for a participant. Therefore, we conducted an offline analysis to investigate how the tracking error affects the PAM identification performance. For the PAM model identification, our method exploits the most informative data point that maximizes the approximated mutual information. To simulate the tracking error, we randomly shifted the recorded data that has the maximum mutual information. The amount of the data time shifts Δ were sampled from a uniform distribution with the range of [-250, 250]corresponding to 1.0 second ahead and delay. We selected the data points $x_{t_{i}^{l}+\Delta}$ and $y_{t_{i}^{l}+\Delta}$. We set this value so that the subjects would show the tracking error within this range. The specified range represents the possible deviation of human subject movements. We used the same number of the collected data and hyperpriors as the actual online experiment introduced in Section 5.1. Figure 9 shows the learning curves of the force prediction errors with the shifted and without-shifted data. In both conditions, the force prediction errors were decreased successfully through the simulated learning process. The results indicate the robustness of our model identification procedure against the tracking error of a subject to the reference trajectories. Meanwhile, the learning curve without the data shift indicates better learning performance. This result indicates that the selected reference trajectory by our active learning method contains more information for PAM model identification than other data points.

6. Discussion

In this section, we further discuss how the hyperprior distribution $p(\theta)$ affects the model identification performance through offline analyses. In addition, we investigate the learned GP-PAM model in this experiment.

6.1. Effects of the hyperpriors

The hyperprior settings of each actuator for the GP hyperparameter learning in our experiments are summarized in Table 1. Even though we manually selected these hyperpriors, the settings of the hyperpriors are not highly sensitive.

Using the dataset we collected for the offline analysis in Section 5.3, we investigated how the model identification performance can be affected by the different settings of the hyperpriors. First, we modulated the hyperpriors for the mean value of the hyperparameters. Instead of using the specific mean values in Table 1, to validate the sensitivity of the value for the performance, we additionally evaluated other parameters locally and randomly sampled at around $m_h = 2.0$ from $\mathcal{N}(2, 1)$ at each experiment. In this offline analysis, we used K = 30 points of the subsampled data to update the hyperparameters at each trial (see Appendix C). We conducted the comparison between the contraction force prediction performances of the fixed hyperprior used in our experiments and that of the stochastically sampled



Fig. 9. Robustness of the model-learning process. The vertical axis shows force prediction error. To simulate tracking error, we randomly shifted recorded data that has maximum mutual information. The amplitude of data shifts were sampled from a uniform distribution. The cyan line shows learning performance with data shift. The blue line shows learning performance without data shift. Results indicate robustness of our model identification procedure against tracking error of participants to reference trajectories. Results also indicate that the reference trajectory selected by our active learning method contains more information for PAM model identification than other data points.

Table 1.	E	lyperprior	settings.
----------	---	------------	-----------

$p(\boldsymbol{\theta}) = \mathcal{N}(m_h, \sigma_h^2)$	PAM1		PAM2		PAM3		PAM4	
	m_h	σ_h	$\overline{m_h}$	σ_h	$\overline{m_h}$	σ_h	$\overline{m_h}$	σ_h
a_1	1.5	10.0	1.8	8.0	2.3	8.0	2.5	8.0
a ₂	2.3	10.0	1.0	8.0	2.6	8.0	2.9	8.0
<i>a</i> ₃	2.2	10.0	2.1	8.0	2.7	8.0	2.0	8.0
σ_k	1.8	10.0	0.7	8.0	2.9	8.0	1.6	8.0
σ_n	5.0	1.4	5.0	1.0	5.0	1.0	5.0	1.0

hyperpriors. We did not find any statistically significant difference between these two performances (Wilcoxon's signed-rank test, 5% significance level). The result suggests that the performance of our method is not very sensitive to the choice in m_h .

Next, we tested much smaller hyperpriors for the variance compared with the settings used in the online experiments. In the online experiments, to avoid overfitting, we used large hyperpriors for the variance. Here, we used smaller hyperpriors to assess the robustness of our proposed method. We conducted the comparison between the contraction force prediction performances of the large hyperpriors for the variance used in our experiments and the newly tested small hyperpriors. We did not find any statistically significant difference between these two performances (Wilcoxon's signed-rank test, 5% significance level). The results indicate that our proposed method is not highly sensitive to the settings of hyperpriors for the variances.

6.2. Learning GP-PAM models

Here, we first investigated which reference trajectory was frequently chosen by our active learning framework. Figure 10 shows the top three most frequently selected reference trajectories. A common tendency can be observed among these trajectory profiles, cf., Figure 5. This result indicates the consistency of the learning process of our proposed approach.

Figure 11 shows the learned GP-PAM model for PAM1. We plotted the function in Figure 11(a) by taking the PAM pressure P and the PAM velocity \dot{L} as the inputs and the contraction force F as the output with the fixed PAM length of L = 240.0 mm that is around the intermediate value of the operated range. The learned function clearly depends on the velocity \dot{L} especially in the high-pressure input region while the static PAM model (see Appendix D) cannot represent velocity-dependent variation of the contraction force. To further investigate the acquired function from a different



Fig. 10. Top three most frequently selected reference trajectories. A common tendency can be observed among the top three trajectory profiles, cf., Figure 5.

view, we also plotted the function in Figure 11(b) by taking the PAM velocity \dot{L} and the PAM length L as the inputs and the contraction force F as the output with the fixed PAM pressure of P = 4.0 bar, which is around the intermediate value of the operated range. Figure 11(a) and (b) indicate that the GP-PAM model successfully captured the nonlinear characteristics of the PAM through our proposed model identification strategy.

7. Conclusion

In this study, we have proposed a design for physical userrobot interactions for modeling soft actuators equipped with wearable robots. We have applied our approach to a userrobot collaborative PAM model identification. We have adopted a GP active learning method for an efficient model identification. We have conducted interactive model identification experiments with two participants using a two-DoF exoskeleton robot containing four PAMs. These experimental results indicate that our proposed GP active learning method with the interactive identification process is more suitable for soft actuator modeling than the standard PAM modeling procedure that did not consider physical userrobot interactions.

We have further investigated how the user's tracking performances based on the presented reference trajectories affected the model identification process through offline analysis. We found that the proposed approach is robust against the user's tracking error. However, if no tracking error is present, the learning performance is better than that of the simulated tracking errors. This indicated that our active learning approach successfully selected informative reference trajectories for model identification to the user.



Fig. 11. Learned GP-PAM model (PAM1): (a) taking PAM pressure *P* and PAM velocity \dot{L} as inputs and contraction force *F* as output with fixed PAM length of L = 240.0 mm, and (b) taking PAM velocity \dot{L} and PAM length *L* as inputs and contraction force *F* as output with fixed PAM pressure of P = 4.0 bar. The GP-PAM model successfully captured the nonlinear characteristics of PAM through our proposed model identification strategy.

Currently, we are using the ordinal GP model for our active learning framework. However, we cannot directly use a large number of samples to train the ordinal GP model because of the heavy computational burden. For future work, we will derive an active learning algorithm for a sparse GP model such that we can utilize information extracted from all the collected data, but can simultaneously reduce the effective number of samples to concisely represent the actuator models for practical implementation.

Recently, novel soft actuators have been developed (Koizumi et al., 2018; Li et al., 2017; Marchese et al., 2015; Robertson and Paik, 2017). They have soft materials and are driven by fluid; our approach could be applied to model identification for wearable robots with soft actuators. In addition, our approach could be employed in model identification for soft manipulators that physically interact with users (Bicchi and Tonietti, 2004; Ohta et al., 2018). Recently, wearable sensing devices that can measure human movements (Mengüç et al., 2014; Michael and Howard,

2018) and stiffness (Yagi et al., 2018) have also been developed. We could combine these kinds of devices and our proposed method in future work.

Funding

This study was supported by "Research and development of technology for enhancing functional recovery of elderly and disabled people based on non-invasive brain imaging and robotic assistive devices," the Commissioned Research of National Institute of Information and Communications Technology (NICT), Japan, by ImPACT Program of Council for Science, Technology and Innovation (Cabinet Office, Government of Japan), by AMED (Grant Number JP18he1902005), and by JSPS KAKENHI (Grant Numbers JP16H06565 and JP17J02979).

Notes

- 1. See https://jp.mathworks.com/matlabcentral/fileexchange/428 85-nearestspd
- 2. See also https://github.com/rmgarnett/

ORCID iD

Masashi Hamaya D https://orcid.org/0000-0003-4189-8219

References

- Asbeck AT, De Rossi SM, Holt KG and Walsh CJ (2015) A biologically inspired soft exosuit for walking assistance. *The International Journal of Robotics Research* 34(6): 744–762.
- Bajcsy A, Losey DP, O'Malley MK and Dragan AD (2017) Learning robot objectives from physical human interaction. In: *Proceedings of the Conference on Robot Learning*, pp. 217–226.
- Bicchi A and Tonietti G (2004) Fast and "soft-arm" tactics [robot arm design]. *IEEE Robotics and Automation Magazine* 11(2): 22–33.
- Cao J, Xie SQ and Das R (2018) MIMO sliding mode controller for gait exoskeleton driven by pneumatic muscles. *IEEE Trans*actions on Control Systems Technology 26(1): 274–281.
- Chou CP and Hannaford B (1996) Measurement and modeling of Mckibben pneumatic artificial muscles. *IEEE Transactions on Robotics and Automation* 12(1): 90–102.
- Ding Y, Kim M, Kuindersma S and Walsh CJ (2018) Human-inthe-loop optimization of hip assistance with a soft exosuit during walking. *Science Robotics* 3(15): eaar5438.
- Dollar AM and Herr H (2008) Lower extremity exoskeletons and active orthoses: Challenges and state-of-the-art. *IEEE Transactions on Robotics* 24(1): 144–158.
- Ferris DP, Czerniecki JM and Hannaford B (2005) An ankle–foot orthosis powered by artificial pneumatic muscles. *Journal of Applied Biomechanics* 21(2): 189–197.
- Furukawa Ji, Noda T, Teramae T and Morimoto J (2017) Human movement modeling to detect biosignal sensor failures for myoelectric assistive robot control. *IEEE Transactions on Robotics* 33(4): 846–857.
- Garnett R, Osborne M and Hennig P (2014) Active learning of linear embeddings for Gaussian processes. In: *Proceedings of the Conference on Uncertainty Artificial Intelligence*, pp. 230–239.
- Ghadirzadeh A, Bütepage J, Maki A, Kragic D and Björkman M (2016) A sensorimotor reinforcement learning framework for

physical human-robot interaction. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2682–2688.

- Gopura R, Bandara D, Kiguchi K and Mann GK (2016) Developments in hardware systems of active upper-limb exoskeleton robots: A review. *Robotics and Autonomous Systems* 75: 203–220.
- Hamaya M, Matsubara T, Noda T, Teramae T and Morimoto J (2017a) Learning assistive strategies for exoskeleton robots from user–robot physical interaction. *Pattern Recognition Letters* 99: 67–76.
- Hamaya M, Matsubara T, Noda T, Teramae T and Morimoto J (2017b) User–robot collaborative excitation for PAM model identification exoskeleton robots. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3063–3068.
- Kapoor A, Grauman K, Urtasun R and Darrell T (2007) Active learning with Gaussian processes for object categorization. In: *Proceedings IEEE International Conference on Computer Vision*, pp. 1–8.
- Khoramshahi M and Billard A (2018) A dynamical system approach to task-adaptation in physical human–robot interaction. *Autonomous Robots*, in press.
- Koizumi S, Kurumaya S, Nabae H, Endo G and Suzumori K (2018) Braiding thin Mckibben muscles to enhance their contracting abilities. *IEEE Robotics and Automation Letters* 3(4): 3240–3246.
- Koller JR, Gates DH, Ferris DP and Remy CD (2016) 'Body-inthe-loop' optimization of assistive robotic devices: A validation study. In: *Proceedings of Robotics: Science and Systems*.
- Lagoda C, Schouten AC, Stienen AH, Hekman EE and van der Kooij H (2010) Design of an electric series elastic actuated joint for robotic gait rehabilitation training. In: *Proceedings IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics*, pp. 21–26.
- Li S, Vogt DM, Rus D and Wood RJ (2017) Fluid-driven origamiinspired artificial muscles. *Proceedings of the National Academy of Sciences* 114(50): 13132–13137.
- Liu Q, Liu A, Meng W, Ai Q and Xie SQ (2017) Hierarchical compliance control of a soft ankle rehabilitation robot actuated by pneumatic muscles. *Frontiers in Neurorobotics* 11: 64.
- Maeda G, Ewerton M, Neumann G, Lioutikov R and Peters J (2017) Phase estimation for fast action recognition and trajectory generation in human–robot collaboration. *The International Journal of Robotics Research* 36(13–14): 1579–1594.
- Marchese AD, Katzschmann RK and Rus D (2015) A recipe for soft fluidic elastomer robots. *Soft Robotics* 2(1): 7–25.
- Mengüç Y, Park YL, Pei H, et al. (2014) Wearable soft sensing suit for human gait measurement. *The International Journal of Robotics Research* 33(14): 1748–1764.
- Merola A, Colacino D, Cosentino C and Amato F (2018) Modelbased tracking control design, implementation of embedded digital controller and testing of a biomechatronic device for robotic rehabilitation. *Mechatronics* 52: 70–77.
- Mghames S, Laghi M, Della Santina C, et al. (2017) Design, control and validation of the variable stiffness exoskeleton flexo. In: *Proceedings IEEE International Conference on Rehabilitation Robotics*, pp. 539–546.
- Michael B and Howard M (2018) Gait reconstruction from motion artefact corrupted fabric-embedded sensors. *IEEE Robotics* and Automation Letters 3(3): 1918–1924.

- Murphy KP (2013) Machine Learning: A Probabilistic Perspective. Cambridge, MA: MIT Press.
- Noda T, Furukawa Ji, Teramae T, Hyon SH and Morimoto J (2013) An electromyogram based force control coordinated in assistive interaction. In: *IEEE International Conference on Robotics and Automation*, pp. 2657–2662.
- Noda T, Teramae T, Ugurlu B and Morimoto J (2014) Development of an upper limb exoskeleton powered via pneumatic electric hybrid actuators with Bowden cable. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3573–3578.
- Ohta P, Valle L, King J, et al. (2018) Design of a lightweight soft robotic arm using pneumatic artificial muscles and inflatable sleeves. *Soft Robotics* 5(2): 204–215.
- Park YL, Chen Br, Pérez-Arancibia NO, et al. (2014) Design and control of a bio-inspired soft wearable robotic device for ankle–foot rehabilitation. *Bioinspiration & Biomimetics* 9(1).
- Peternel L, Noda T, Petric T, Ude A, Morimoto J and Babic J (2016a) Adaptive control of exoskeleton robots for periodic assistive behaviours based on emg feedback minimisation. *PLoS ONE* 11(2): 1–26.
- Peternel L, Tsagarakis N and Ajoudani A (2016b) Towards multimodal intention interfaces for human–robot co-manipulation. In: Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2663–2669.
- Rasmussen C and Williams C (2006) Gaussian Processes for Machine Learning. Cambridge, MA: MIT Press.
- Robertson MA and Paik J (2017) New soft robots really suck: Vacuum-powered systems empower diverse capabilities. *Science Robotics* 2(9): eaan6357.
- Rozo L, Calinon S, Caldwell DG, Jimenez P and Torras C (2016) Learning physical collaborative robot behaviors from human demonstrations. *IEEE Transactions on Robotics* 32(3): 513–527.
- Snelson E and Ghahramani Z (2006) Sparse Gaussian processes using pseudo-inputs. Advances in Neural Information Processing Systems 18: 1257.
- Tanaka D, Matsubara T, Ichien K and Sugimoto K (2014) Object manifold learning with action features for active tactile object recognition. In: Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 608–614.
- Teramae T, Noda T, Hyon SH and Morimoto J (2013) Modeling and control of a pneumatic–electric hybrid system. In: Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4887–4892.
- Teramae T, Noda T and Morimoto J (2018) EMG-based model predictive control for physical human–robot interaction: Application for assist-as-needed control. *IEEE Robotics and Automation Letters* 3(1): 210–217.
- Thatte N, Duan H and Geyer H (2017) A sample-efficient blackbox optimizer to train policies for human-in-the-loop systems with user preferences. *IEEE Robotics and Automation Letters* 2(2): 993–1000.
- Ugurlu B, Doppmann C, Hamaya M, et al. (2016) Variable ankle stiffness improves balance control: Experiments on a bipedal exoskeleton. *IEEE Transactions on Mechatronics* 21(1): 79–87.
- Ugurlu B, Forni P, Doppmann C and Morimoto J (2015) Torque and variable stiffness control for antagonistically driven pneumatic muscle actuators via a stable force feedback controller. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1633–1639.

- Wang S, Wang L, Meijneke C, et al. (2015) Design and control of the mindwalker exoskeleton. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 23(2): 277–286.
- Yagi K, Suzuki K and Mochiyama H (2018) Human joint impedance estimation with a new wearable device utilizing snapthrough buckling of closed-elastica. *IEEE Robotics and Automation Letters* 3(3): 1506–1513.
- Yan T, Cempini M, Oddo CM and Vitiello N (2015) Review of assistive strategies in powered lower-limb orthoses and exoskeletons. *Robotics and Autonomous Systems* 64: 120–136.
- Yu H, Huang S, Chen G, Pan Y and Guo Z (2015) Human-robot interaction control of rehabilitation robots with series elastic actuators. *IEEE Transactions on Robotics* 31(5): 1089–1100.
- Zhang J, Fiers P, Witte KA, et al. (2017) Human-in-the-loop optimization of exoskeleton assistance during walking. *Science* 356(6344): 1280–1284.

Appendix A. MAP estimation of the GP hyperparameter

The hyperparameter can be obtained from the *maximum a* posteriori of $p(\theta|X, y) \propto p(y|X, \theta)p(\theta)$. The hyperprior model $p(\theta)$ is provided in Table 1. We consider minimizing the negative log likelihood:

$$\log (p(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{y})) = \frac{1}{2} \boldsymbol{y}^{\mathrm{T}} (\boldsymbol{K} + \sigma_{n}^{2} \mathbf{I})^{-1} \boldsymbol{y} + \frac{1}{2} \log |\boldsymbol{K} + \sigma_{n}^{2} \mathbf{I}| + \frac{n}{2} \log 2\pi - \log p(\boldsymbol{\theta})$$
(10)

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \{ -\log\left(p(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{y})\right) \}$$
(11)

The gradient of $\log(p(\theta|X, y))$ can be calculated analytically. We used the conjugate gradient method to update $\hat{\theta}$.

Appendix B. Approximated marginal predictive variance

The marginalization σ_f^2 can be given as

$$\sigma_{f}^{2}(\mathbf{x}) = \frac{4}{3}\sigma_{f|\theta}^{2}(\mathbf{x}) + \frac{\partial m_{f|\theta}(\mathbf{x})}{\partial \theta}^{\mathrm{T}} \Sigma \frac{\partial m_{f|\theta}(\mathbf{x})}{\partial \theta} + \frac{1}{3\sigma_{f|\theta}^{2}(\mathbf{x})} \frac{\partial \sigma_{f|\theta}^{2}(\mathbf{x})}{\partial \theta}^{\mathrm{T}} \Sigma \frac{\partial \sigma_{f|\theta}^{2}(\mathbf{x})}{\partial \theta}$$
(12)

where $m_{f|\theta}(\mathbf{x})$ and $\sigma_{f|\theta}^2(\mathbf{x})$ are the predictive mean and variance of the distribution $p(f(\mathbf{x})|\mathbf{X}, \mathbf{y}, \boldsymbol{\theta})$. Partial derivatives with respect to the hyperparameters $\boldsymbol{\theta}$ are derived around $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$. Here Σ is the covariance matrix of $p(\boldsymbol{\theta}|\mathbf{X}, \mathbf{y})$ that is obtained by Laplace approximation:

$$\Sigma^{-1} = -\nabla \nabla^{\mathrm{T}} \log \left(p(\boldsymbol{\theta} | \boldsymbol{X}, \boldsymbol{y}) \right) |_{\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}}$$
(13)

Appendix C. Data subsampling

As the computational complexity of GP active learning is $O(n^2)$, the computational cost increases significantly with

a large number of training data points. To cope with this issue, we subsampled data points from the pair of the acquired input X_s and output y_s data with considering the mutual information between the function f and the hyperparameters θ . Concretely, we first extract the most informative data point in terms of the mutual information:

$$t_1^{\mathrm{I}} = \arg\max_t \{I(\boldsymbol{x}_t^s)\}$$
(14)

where $X_s = [x_1^s, ..., x_T^s]$. We then additionally subsampled data points by randomly selected data indexes $\{t_2^I, ..., t_K^I\}$. The number of subsampled data points were K = 3 for our online experiments. As the pair of the input X_s and output y_s data are acquired through the informative reference trajectory as explained in Algorithm 1, here we assume that data points in X_s and y_s are equivalently informative to estimate the hyperparameter θ except for the most informative data point t_1^I . By using the above subsampling strategy, we have a reduced number of data points $\{x_{t_1^I}, ..., x_{t_K^I}\}$ and $\{y_{t_1^I}, ..., y_{t_K^I}\}$. We add these subsampled data points to the input X and output y dataset.

Appendix D. Parametric models

We developed our pneumatic actuators based on FESTO PAMs (see https://www.festo.com). FESTO provided a data sheet that shows the relationship between the contraction rate *C* and force output *F*. From Chou and Hannaford (1996) and the data sheet provided by FESTO, we modeled the relationship between the contraction rate and the force output by a quadratic function $f = \alpha_1 C^2 + \alpha_2 C + \alpha_3$ (see Figure 12). We identified the two sets of parameters for the quadratic models $\{\alpha_1^l, \alpha_2^l, \alpha_3^l\}$ and $\{\alpha_1^u, \alpha_2^u, \alpha_3^u\}$ under two different pressure inputs to PAM. Then, force outputs were estimated by using a linear approximation of the data sheet profiles Teramae et al. (2013) as

$$F = \frac{(f_u - f_l)P + P_u f_l - P_l f_u}{P_u - P_l}$$
(15)

$$f_l = \alpha_1^l C^2 + \alpha_2^l C + \alpha_3^l \tag{16}$$

$$f_u = \alpha_1^u C^2 + \alpha_2^u C + \alpha_3^u \tag{17}$$

where f_u and f_l are the contraction force at the upper and lower pressures, respectively. In this study, we designed $P_u = 6.0$ bar and $P_l = 2.0$ bar, but these upper and lower



Fig. 12. Static relations between the contraction rate and maximum contraction force. We used the data sheet provided by FESTO as a reference.

pressure value can be arbitrarily selected as long as these two values are not very close and within a reasonable range for the target PAMs.

Appendix E. Sparse GP

Snelson and Ghahramani (2006) proposed the SPGP that is parameterized by the pseudo-inputs l ($l \ll n$). The SPGP can reduce the computational cost ($\mathcal{O}(ln^2)$ for training and $\mathcal{O}(l^2)$ for prediction) compared with the normal GP ($\mathcal{O}(n^3)$ for training and $\mathcal{O}(n^2)$ for prediction). Given pseudo-inputs $\bar{\mathbf{X}} = [\bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_l]$, the predictive distribution of the SPGP can be expressed as follows:

$$p(y_*|\mathbf{x}_*, \mathcal{D}, \bar{\mathbf{X}}) = \mathcal{N}(y_*|\boldsymbol{\mu}_*, \boldsymbol{\sigma}_*^2)$$
(18)

where

$$\boldsymbol{\mu}_{*} = \mathbb{E}[\boldsymbol{y}_{*}] = \boldsymbol{k}_{*}^{\mathrm{T}} \boldsymbol{Q}_{ln}^{-1} (\boldsymbol{K}_{ln} + \sigma_{n}^{2} \mathbf{I})^{-1} \boldsymbol{y}$$
(19)

$$\sigma_*^2 = \operatorname{Var}[y_*] = k_{**} - k_*^{\mathrm{T}} (K_l - \mathbf{Q}_l^{-1}) k_* \qquad (20)$$

$$\mathbf{Q}_l = \mathbf{K}_l + \mathbf{K}_{ln}(\mathbf{\Lambda} + \sigma_n^2 \mathbf{I})$$
(21)

The marginal likelihood can be maximized with respect to hyperparameters (Λ and σ_n) and pseudo-inputs $\bar{\mathbf{X}}$ by a gradient method.