2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Meta-Learning-Based Optimal Control for Soft Robotic Manipulators to Interact with Unknown Environments

Zhiqiang Tang[1], Peiyi Wang[2], Wenci Xin[1], Zhexin Xie[1],
Longxin Kan[1], Muralidharan Mohanakrishnan[1], Cecilia Laschi[1]

***Abstract*— Safe and efficient robot-environment interaction is a critical but challenging problem as robots are being increasingly employed to operate in unstructured and unpredictable environments. Soft robots are inherently compliant to safely interact with environments but their high nonlinearity exacerbates control difficulties. Meta-learning provides a powerful tool for fast online model adaptation because it can learn an efficient model from data across different environments. Thus, this work applies the idea of meta-learning for the control of soft robotics. In particular, a target-oriented proactive search strategy is firstly performed to collect environment-specific data efficiently when a new interaction environment occurs. Then meta-learning exploits past experience to train a data-driven probabilistic model prior, and the model prior is online updated to be fast adapted to the new environment. Lastly, a model-based optimal control policy is utilized to drive the robot to desired performance. Our approach controls a soft robotic manipulator to achieve the desired position and contact force simultaneously when interacting with unknown changing environments. Overall, this work provides a viable control approach for soft robots to interact with unknown environments.**

***Index Terms*— Modeling, Control, and Learning for Soft Robots, Physical Human-Robot Interaction, Force Control**

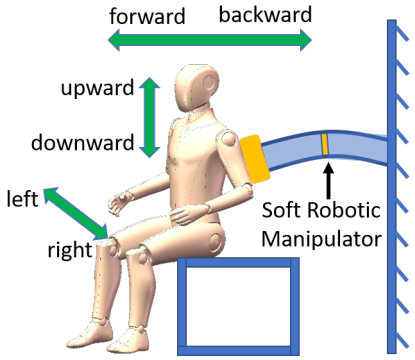


Fig. 1: A schematic diagram of a soft robotic manipulator assisting people in a showering activity. During the showering, people may have uncertain motions, such as forward-backward, upward-downward, and left-right movements. The soft robotic manipulator should be able to achieve desired performance during such unpredictable interaction environments.

I. Introduction

Physical interactions between robots and environments/humans are challenging problems due to diverse environmental conditions as well as uncertain human motions [1], [2]. Various field robots are required to deal with complex environments, for instance, aerial robots may fly through highly cluttered environments such as dense forests [3], legged robots need to overcome diverse terrains to achieve dynamic locomotion in natural environments [4]. Besides, robots in human daily lives are expected to interact with humans safely and effectively. Uncertainty arises if social robots have difficulties in accurately interpreting human's social behavior [5], assistive robots lack critical information for carrying out daily tasks [6]. Modern robots need to operate in increasingly unstructured environments that are inherently unpredictable, which requires robots being adaptive to unforeseen environments.

The objective of this study is to develop a control approach such that the robots can achieve desired interactions with prior unknown environments. Traditional adaptive control approaches usually require expert knowledge to derive accurate dynamics models and constraints of both robots and environments [7], which are difficult to know beforehand in unforeseeable environments. In recent years, meta-learning becomes popular to deal with dynamic environments. Meta-learning, or "learning to learn", is a learning principle that exploits past experiences of the robot in various environments for fast adaptation to a similar environment with only a few observations [8]–[10]. Meta-learning is based on the assumption that all environments share common structures. Researchers in [11] presented a model-based meta-reinforcement learning algorithm to control a legged millirobot under different situations. Meanwhile, authors in [12], [13] developed meta-learning control policies combined with situation embeddings to enable quadrupedal robots being fast adaptive to new environments. Moreover, researchers in [14] presented an adaptive-control-oriented meta-learning method while authors in [15] proposed a domain adversarially invariant meta-learning approach to enable fast online adaptation of uninhabited aerial vehicles in different wind speeds. These recent studies show promising directions for robots in unknown environments and inspire us to apply meta-learning in our soft robotic applications.

In this study, we consider a particular soft robotic application where a soft robotic manipulator assists elderly

This research is supported by the National Research Foundation, Singapore, under its Medium Sized Centre Programme-Centre for Advanced Robotics Technology Innovation (CARTIN) and by the National Robotics Program, Singapore, under the Soft & Hybrid Robotics Phase 2a project.

[1] Department of Mechanical Engineering, National University of Singapore, Singapore (email: zq.tang@nus.edu.sg; mpeclc@nus.edu.sg)
[2] Robotics Research Center, Beijing Jiaotong University, Beijing, China.

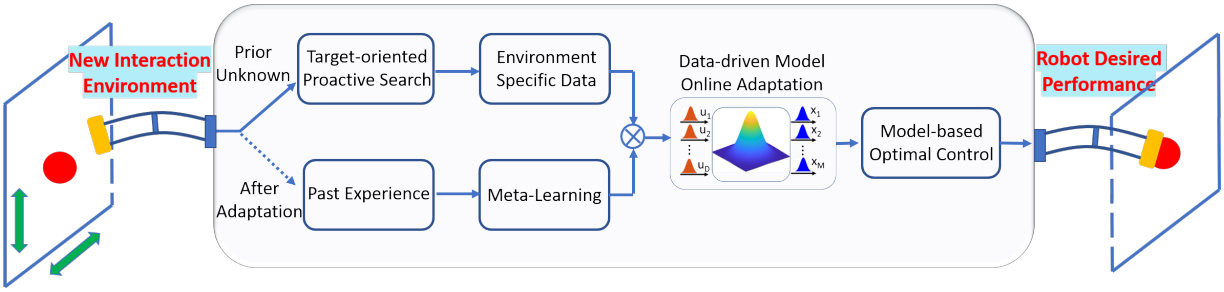


Fig. 2: Overview of the meta-learning-based optimal control approach. The input is a new interaction environment and the output is desired robot performance. The target-oriented proactive search strategy aims to collect efficient environment-specific data with minimal interaction trials while providing the best-suited data for model adaptation. Meanwhile, meta-learning exploits past experience to train a data-driven model prior such that the model can be online fast adapted to the new environment. Afterwards, model-based optimal control strategy is utilized to drive the robot to desired performance. Finally, new environmental data is recorded into the experience dataset after adaptation.

people in a showering activity, as shown in Fig. 1. Elderly people need assistance in their daily tasks due to age-related functional loss while showering is one of the most critical but high risk task [16]. The soft robotic manipulator exploits the principle of soft robotic technologies to ensure safe and comfortable human-robot interactions during the intimate showering activity [17]. In order to successfully assist the showering task, it would require simultaneous position and force control for the soft robotic manipulator [18]. However, humans may have uncertain motions during showering, such as forward-backward, upward-downward, and left-right movements. This causes big challenges for the control of soft robotic manipulator because of unpredictable interaction environments. Furthermore, nonlinear properties of soft robots and system uncertainties make it a difficult task to acquire accurate models. Previous hybrid position/force control methods [19]–[21] usually require analytical models that are difficult to derive for soft robotic manipulators without making significant simplifications. Moreover, most control strategies developed for soft robots are limited in free space without constrains or interactions with static structured environments [22], [23].

To achieve our target that the soft robotic manipulator can have desired interactions with unknown environments, we propose a meta-learning-based optimal control approach combined with a data-driven model. The basic idea of our approach is described in Fig. 2, which consists of three key components, i.e. target-oriented proactive search, meta-learning with the online adaptation of a data-driven model, and model-based optimal control. First, the target-oriented proactive search aims to collect environment-specific data efficiently to minimize interaction trials. Different from passive learning [24] or random search [25] that may take unnecessary actions, the proactive search is based on the idea of active learning [26] that robots will take purposeful actions to gather effective data when facing a prior unknown environment. These effective data best realize a learning objective. A recent study in [27] also uses proactive online planning to successfully detect unsafe situations for drones. Our proactive search strategy is built upon Bayesian optimization to find optimal control actions because Bayesian optimization is an efficient approach for finding optimal values of expensive cost functions [28]. Additionally, a data-driven probabilistic model is established to represent system dynamics, which explicitly captures system nonlinearity and uncertainty. Meta-learning exploits history data information for fast online model adaptation. Previous data-driven methods [18], [29] that controlled continuum/soft manipulators to achieve position and force control were conducted in static unchanging environments. These methods heavily relied on environment-specific data and thus would have poor performance when the environment changes. Finally, optimal control actions are computed by minimizing a cost function based on the learned model.

Major contributions of this study include:

- developing a meta-learning-based optimal control approach for soft robots;
- controlling the robot's position and contact force simultaneously in uncertain changing environments;
- presenting a target-oriented proactive search strategy to minimize interaction trials.

The rest of this article is organized as follows: Section II introduces a data-driven probabilistic model and meta-learning to train model parameters; Section III presents the details of target-oriented proactive search strategy and model-based optimal control policy; Section IV demonstrates experimental results of a soft robotic manipulator; Finally, Section V concludes this article with a summary of achievements and discussion for future developments.

II. Model Learning

In this section we present a data-driven method to model the system dynamics and a meta-learning algorithm to train the model parameters.

A. Data-driven Model

We model the soft robotic system using multi-task Gaussian Process (GP) [30]. The multi-task GP can describe a

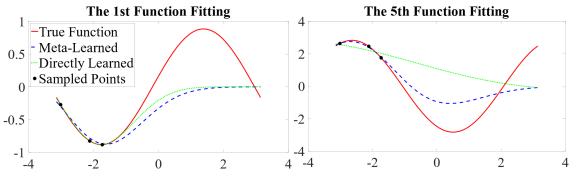


Fig. 3: The performance comparison between meta-learning and direct learning in terms of a simple 1-dimensional sine wave. The amplitude and phase of the sine wave are randomly changed. Direct learning only uses the current sample points while meta-learning also exploits previous function information. Meta-learning obviously outperforms direct learning after 5 function changes.

Algorithm 1 Meta-learning for Model Parameters

1: **Input:** Already collected J environments, randomly initialize $\boldsymbol{\theta}$, new environment data.
2: **Output:** Learned model parameters $\boldsymbol{\theta}^*$ for the new environment.
3: **for** iteration = 1, 2, $\cdots$, J **do**
4: Sample date from collected environments
5: Compute the gradient of loss function in Eq. (5)
6: Calculate model parameters $\boldsymbol{\theta}'$ in Eq. (6) by m steps
7: Update $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha(\boldsymbol{\theta}' - \boldsymbol{\theta})$
8: **end for**
9: Train on new environment data by steps 5 to 7.

multi-input multi-output system and its probabilistic property can explicitly capture system uncertainties as well as correlations among outputs. In our robotic system, state $\boldsymbol{x}_{t+1}$ at time $t+1$ depends on both state $\boldsymbol{x}_t$ at time t and action $\boldsymbol{u}_t$ as well as sensor observation noise. This relationship is described as follows:

$$\boldsymbol{x}_{t+1} = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t) + \boldsymbol{w} \tag{1}$$

with state variables $\boldsymbol{x}_t \in \mathbb{R}^{M\times 1}$, control actions $\boldsymbol{u}_t \in \mathbb{R}^{D\times 1}$, and i.i.d. Gaussian system noise $\boldsymbol{w} \in \mathbb{R}^{M\times 1}$, each $w_i \sim \mathcal{N}(0,\ \sigma_{wi}^2)$, and the transition dynamics $\boldsymbol{f}$. The Gaussian process f can be completely determined by its mean function and covariance function. To learn the Gaussian process from data, we use tuples $\boldsymbol{v}_t = [\boldsymbol{x}_t, \boldsymbol{u}_t] \in \mathbb{R}^{(M+D)\times 1}$ as training inputs and $\boldsymbol{y}_t = \boldsymbol{x}_{t+1}$ as training outputs. In this study, we assume that the GPs have zero mean. We directly induce correlations between outputs by using a positive semi-definite matrix $\boldsymbol{K}^f \in \mathbb{R}^{M\times M}$ and a squared exponential covariance over training inputs:

$$k^v(\boldsymbol{v}_i,\ \boldsymbol{v}_j) = \sigma_f^2 \exp\big(-\frac{1}{2}(\boldsymbol{v}_i - \boldsymbol{v}_j)^T \boldsymbol{L}^{-1}(\boldsymbol{v}_i - \boldsymbol{v}_j)\big) \tag{2}$$

where σ_f^2 is the signal variance, $\boldsymbol{L} = diag([l_1^2, \cdots, l_{M+D}^2]) \in \mathbb{R}^{(M+D)\times(M+D)}$ is characteristic length-scale. Given N training inputs $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_N] \in \mathbb{R}^{N\times(D+M)}$ and corresponding observations $\boldsymbol{Y} = vec[\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_N]$ which is the vector of function values corresponding to $\boldsymbol{y}$, i.e. $\boldsymbol{Y} = (\boldsymbol{y}_{11}, \cdots, \boldsymbol{y}_{N1}, \cdots, \boldsymbol{y}_{12}, \cdots, \boldsymbol{y}_{N2}, \boldsymbol{y}_{1M}, \cdots, \boldsymbol{y}_{NM})^T \in \mathbb{R}^{MN\times 1}$, the predicted state at test point $\boldsymbol{v}_t = [\boldsymbol{x}_t, \boldsymbol{u}_t]$ is Gaussian distributed with the mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\Sigma}$ as follows:

$$\begin{aligned}
&\mathcal{P}(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_t) = \boldsymbol{\mathcal{N}}(\boldsymbol{x}_{t+1}|\boldsymbol{\mu}_{t+1},\ \boldsymbol{\Sigma}_{t+1}) \\
&\boldsymbol{\mu}_{t+1} = (\boldsymbol{K}^f \otimes \boldsymbol{k}^*)^T \boldsymbol{K}^{-1} \boldsymbol{Y} \\
&\boldsymbol{\Sigma}_{t+1} = (\boldsymbol{K}^f \otimes k^{**}) - (\boldsymbol{K}^f \otimes \boldsymbol{k}^*)^T \boldsymbol{K}^{-1} (\boldsymbol{K}^f \otimes \boldsymbol{k}^*) \\
&\boldsymbol{K} = \boldsymbol{K}^f \otimes \boldsymbol{K}^v + \boldsymbol{\Sigma}_w \otimes \boldsymbol{I}
\end{aligned} \tag{3}$$

where $\otimes$ denotes the Kronecker product. $\boldsymbol{K}^v \in \mathbb{R}^{N\times N}$ is the matrix of covariance between all pairs of training points, i.e. $K_{ij}^v = k^v(\boldsymbol{v}_i,\ \boldsymbol{v}_j)$, $\boldsymbol{\Sigma}_w \in \mathbb{R}^{M\times M}$ is a diagonal matrix in which the (i,i)th element is σ_{wi}^2, $\boldsymbol{I}$ is an identity matrix with proper dimension. $\boldsymbol{k}^* = [k^v(\boldsymbol{v}_t,\ \boldsymbol{v}_1),\ k^v(\boldsymbol{v}_t,\ \boldsymbol{v}_2), \cdots, k^v(\boldsymbol{v}_t,\ \boldsymbol{v}_N)] \in \mathbb{R}^{N\times 1}$ is the vector of covariance between the test point $\boldsymbol{v}_t$ and all the training points $\boldsymbol{V}$, $k^{**} = k^v(\boldsymbol{v}_t,\ \boldsymbol{v}_t) \in \mathbb{R}$ is covariance of test point itself.

B. Meta-learning for Model Parameters

A gradient-based meta-learning algorithm [8], [9] is designed to train hyper-parameters (i.e. $\Big(\boldsymbol{\theta} = \{\sigma_f,\ \sigma_{wi},\ l_i,\ K_{ij}^f\}\Big)$ of the Gaussian process model. The meta-learning algorithm assumes that all environments (including both trained and unknown ones) are drawn from the same distribution and share a common structure that can be exploited for fast learning. The goal of this meta-learning algorithm is to find a proper initial model parameters $\boldsymbol{\theta}$ based on a loss function such that the learned model parameters $\boldsymbol{\theta}'$ can be quickly adapted to the new environment by taking only a few gradient descent steps from the initial parameters.

The loss function of meta-learning is the negative log-likelihood of data under the Gaussian process model:

$$\begin{aligned}
L(\boldsymbol{\theta}) &= -\log \mathcal{P}(\boldsymbol{Y}|\boldsymbol{V}, \boldsymbol{\theta}) \\
&= \frac{1}{2}\left[\boldsymbol{Y}^T \boldsymbol{K}^{-1} \boldsymbol{Y} + \log|\boldsymbol{K}| + N\log 2\pi\right]
\end{aligned} \tag{4}$$

According to [31], the gradient of the loss function with respect to model parameters can be computed as follows:

$$\begin{aligned}
\frac{\partial L}{\partial \theta_j} &= -\frac{1}{2}\boldsymbol{Y}^T \boldsymbol{K}^{-1} \frac{\partial \boldsymbol{K}}{\partial \theta_j} \boldsymbol{K}^{-1} \boldsymbol{Y} + \frac{1}{2}\operatorname{tr}\left(\boldsymbol{K}^{-1}\frac{\partial \boldsymbol{K}}{\partial \theta_j}\right) \\
&= \frac{1}{2}\operatorname{tr}\left(\left(\boldsymbol{K}^{-1} - \boldsymbol{K}^{-1}\boldsymbol{Y}(\boldsymbol{K}^{-1}\boldsymbol{Y})^T\right)\frac{\partial \boldsymbol{K}}{\partial \theta_j}\right)
\end{aligned} \tag{5}$$

Then the model parameters are updated by gradient descent:

$$\boldsymbol{\theta}' = \boldsymbol{\theta} - \alpha \frac{\partial L(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \tag{6}$$

where α is the step size. The procedure of meta-learning is described in Algorithm 1. Firstly, the model parameters are randomly initialized. Then we take sampled data from previously collected environments. Afterwards, the parameters are updated according to these sampled data using gradient descent. Lastly, the meta-learned parameters are trained to be adaptive in the new environment. Figure 3 shows a simple demonstration of a one-dimensional sine wave. The sine

wave changes its amplitude and phase randomly from the same distribution. When compared with direct learning that only uses current sample points, meta-learning shows better performance because of the learned meta knowledge.

III. Control Policy

During the model parameter learning process, new environment data needs to be collected for training environment-specific parameters. Instead of random collections, we use target-oriented proactive search strategy to improve interaction efficiency. After the learned parameters, model-based optimal control is applied to achieve desired performance. The details of our control policy are described in the following subsections.

A. Target-oriented Proactive Search

The target-oriented proactive search is based on the theory of Bayesian optimization. Bayesian optimization is one of the most efficient approaches to find optimum values of an objective function with a minimal number of function evaluations required [28]. The objective function can be non-convex without closed-form expression, as long as observations can be obtained at sampled actions [32]. The objective function during proactive search is defined as:

$$G(\boldsymbol{x}_t) = \exp\big(-\frac{1}{2}(\boldsymbol{x}_t - \boldsymbol{x}^*)^T\boldsymbol{\Lambda}^{-1}(\boldsymbol{x}_t - \boldsymbol{x}^*)\big) \tag{7}$$

where $\boldsymbol{x}^*$ is the target state and $\boldsymbol{\Lambda}$ is a weight matrix. Since the system model is unknown in a new environment, we assume a commonly used Gaussian process prior over the objective function. Then a surrogate model g can be derived as a Gaussian process regression. Given T inputs $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_T]$ and corresponding outputs $\boldsymbol{G} = [G_1, G_2, \cdots, G_T]$, the posterior distribution of g at a test point $\boldsymbol{u}$ is as follows:

$$\begin{aligned} g(\boldsymbol{u}) &\sim \mathcal{GP}(\mu(\boldsymbol{u}), \sigma^2(\boldsymbol{u})) \\ \mu(\boldsymbol{u}) = \boldsymbol{k}^T\boldsymbol{K}^{-1}\boldsymbol{G} &\quad \sigma^2(\boldsymbol{u}) = k(\boldsymbol{u},\boldsymbol{u}) - \boldsymbol{k}^T\boldsymbol{K}^{-1}\boldsymbol{k} \end{aligned} \tag{8}$$

where k is a squared exponential kernel, $\boldsymbol{k}$ is kernel vector ($\boldsymbol{k}_i = k(\boldsymbol{u}, \boldsymbol{u}_i)$) and $\boldsymbol{K}$ is the kernel matrix ($\boldsymbol{K}_{ij} = k(\boldsymbol{u}_i, \boldsymbol{u}_j)$). This GP model can be learned by evidence maximization [31].

Based on the surrogate model, an acquisition function is designed to guide the search for the optimum. We can select an control action that has a high probability near optimum by maximizing the acquisition function. Here, expected improvement (EI) is utilized as our acquisition function because EI balances the amount of improvement and the probability of improvement [33]. The expression of EI is given by:

$$\begin{aligned} \mathrm{EI}(\boldsymbol{u}) &= \int_{g^*}^{\infty}(g(\boldsymbol{u}) - g^*)\mathcal{N}\left(\mu(\boldsymbol{u}), \sigma^2(\boldsymbol{u})\right) dg(\boldsymbol{u}) \\ &= (\mu(\boldsymbol{u}) - g^*)\Phi(\frac{\mu(\boldsymbol{u}) - g^*}{\sigma(\boldsymbol{u})}) + \sigma(\boldsymbol{u})\phi(\frac{\mu(\boldsymbol{u}) - g^*}{\sigma(\boldsymbol{u})}) \end{aligned} \tag{9}$$

where g^* is the currently observed maximum G value, Φ is normal cumulative distribution function, and ϕ is normal probability density function.

B. Model-based Optimal control

After the target-oriented proactive search, environment-specific data are collected and model parameters are learned to adapt to the new environment. Based on the learned model, the optimal control actions $\boldsymbol{u}_t^*$ are computed by minimizing a cost function $C(\boldsymbol{x}_t) = 1 - G(\boldsymbol{x}_t)$ ($G(\boldsymbol{x}_t)$ is in equation 7). The expression is given by:

$$\begin{aligned} \boldsymbol{u}_t^* &= \arg\min_{\boldsymbol{u}_t} \mathbb{E}[C(\boldsymbol{x}_t)] \\ \mathbb{E}[C(\boldsymbol{x}_t)] &= \int C(\boldsymbol{x}_t)\mathcal{P}(\boldsymbol{x}_t)d\boldsymbol{x}_t = 1 - R \\ R &= \int \exp\big(-\frac{1}{2}(\boldsymbol{x}_t - \boldsymbol{x}^*)^T\boldsymbol{\Lambda}^{-1}(\boldsymbol{x}_t - \boldsymbol{x}^*)\big)\mathcal{P}(\boldsymbol{x}_t)d\boldsymbol{x}_t \end{aligned} \tag{10}$$

From equation (3) we know that $\mathcal{P}(\boldsymbol{x}_t) \sim \boldsymbol{\mathcal{N}}(\boldsymbol{\mu}_t,\ \boldsymbol{\Sigma}_t)$. Then R can be computed as the product of two Gaussian distributions:

$$\begin{aligned} R &= \sqrt{|2\pi\boldsymbol{\Lambda}|}\int \mathcal{H}(\boldsymbol{x}_t)\mathcal{P}(\boldsymbol{x}_t)d\boldsymbol{x}_t \\ \mathcal{H}(\boldsymbol{x}_t) &\sim \boldsymbol{\mathcal{N}}(\boldsymbol{x}^*,\ \boldsymbol{\Lambda}) \quad \mathcal{P}(\boldsymbol{x}_t) \sim \boldsymbol{\mathcal{N}}(\boldsymbol{\mu}_t,\ \boldsymbol{\Sigma}_t) \end{aligned} \tag{11}$$

Based on the production rule of two Gaussians [34], R can be further computed as follows:

$$\begin{aligned} R &= |\boldsymbol{H}|^{1/2}|\boldsymbol{\Sigma}_t|^{-1/2}Z \\ Z &= \exp\big[-\frac{1}{2}\big(\boldsymbol{x}^{*T}\boldsymbol{\Lambda}^{-1}\boldsymbol{x}^* + \boldsymbol{\mu}_t^T\boldsymbol{\Sigma}_t^{-1}\boldsymbol{\mu}_t - \boldsymbol{h}^T\boldsymbol{H}^{-1}\boldsymbol{h}\big)\big] \\ \boldsymbol{H} &= (\boldsymbol{\Lambda}^{-1} + \boldsymbol{\Sigma}_t^{-1})^{-1} \\ \boldsymbol{h} &= \boldsymbol{H}(\boldsymbol{\Lambda}^{-1}\boldsymbol{x}^* + \boldsymbol{\Sigma}_t^{-1}\boldsymbol{\mu}_t) \end{aligned} \tag{12}$$

By simplifying equation (12), we can finally get the expected cost value as follows:

$$\begin{aligned} \mathbb{E}[C(\boldsymbol{x}_t)] =& 1 - |\boldsymbol{I} + \boldsymbol{\Sigma}_t\boldsymbol{\Lambda}^{-1}|^{-\frac{1}{2}} \\ &\times \exp\big[-\frac{1}{2}(\boldsymbol{\mu}_t - \boldsymbol{x}^*)^T(\boldsymbol{\Sigma}_t + \boldsymbol{\Lambda})^{-1}(\boldsymbol{\mu}_t - \boldsymbol{x}^*)\big] \end{aligned} \tag{13}$$

we compute the gradient of the expected cost value $\mathbb{E}[C(\boldsymbol{x}_t)]$ with respect to control action $\boldsymbol{u}_t$ to find the solution. According to the chain rule of derivative, we can compute the gradient as follows:

$$\frac{\partial\mathbb{E}[C(\boldsymbol{x}_t)]}{\partial\boldsymbol{u}_t} = \frac{\partial\mathbb{E}[C(\boldsymbol{x}_t)]}{\partial\boldsymbol{\mu}_t}\frac{\partial\boldsymbol{\mu}_t}{\partial\boldsymbol{u}_t} + \frac{\partial\mathbb{E}[C(\boldsymbol{x}_t)]}{\partial\boldsymbol{\Sigma}_t}\frac{\partial\boldsymbol{\Sigma}_t}{\partial\boldsymbol{u}_t} \tag{14}$$

The partial derivatives $\partial\mathbb{E}[C(\boldsymbol{x}_t)]/\partial\boldsymbol{\mu}_t$, $\partial\mathbb{E}[C(\boldsymbol{x}_t)]/\partial\boldsymbol{\Sigma}_t$ of the expected cost with respect to the mean and the variance of the state distribution $\mathcal{P}(\boldsymbol{x}_t) \sim \boldsymbol{\mathcal{N}}(\boldsymbol{\mu}_t,\ \boldsymbol{\Sigma}_t)$ are needed.

$$\begin{aligned} \frac{\partial\mathbb{E}[C(\boldsymbol{x}_t)]}{\partial\boldsymbol{\mu}_t} &= \mathbb{E}[C(\boldsymbol{x}_t)] \times (\boldsymbol{\mu}_t - \boldsymbol{x}^*)^T(\boldsymbol{\Sigma}_t + \boldsymbol{\Lambda})^{-1} \\ \frac{\partial\mathbb{E}[C(\boldsymbol{x}_t)]}{\partial\boldsymbol{\Sigma}_t} &= -\frac{1}{2}\mathbb{E}[C(\boldsymbol{x}_t)] \times \Big\{(\boldsymbol{\Sigma}_t + \boldsymbol{\Lambda})^{-1} \\ &\qquad (\boldsymbol{\mu}_t - \boldsymbol{x}^*)(\boldsymbol{\mu}_t - \boldsymbol{x}^*)^T - \boldsymbol{I}\Big\}(\boldsymbol{\Sigma}_t + \boldsymbol{\Lambda})^{-1} \end{aligned} \tag{15}$$

Algorithm 2 Meta-Learning-Based Optimal Control

Input: Meta-learned initial model parameters
Input: Encounter a new environment
Output: Desired control performance
repeat
    Perform target-oriented proactive search for T trials
    Collect environment-specific dataset
    Train model parameters on the environment-specific data from the meta-learned initial parameters
    Perform model-based optimal control
until Desired performance achieved
Record the new environment dataset into history data

Then we continue to compute the derivatives of mean and variance with respect to control action.

$$\begin{aligned}\frac{\partial \boldsymbol{\mu}_t}{\partial \boldsymbol{u}_t} &= (\boldsymbol{K}^f \otimes \frac{\partial \boldsymbol{k}^*}{\partial \boldsymbol{u}_t})^T \boldsymbol{K}^{-1}\boldsymbol{Y} \\ \frac{\partial \boldsymbol{\Sigma}_t}{\partial \boldsymbol{u}_t} &= -2\big(\boldsymbol{K}^{-1}(\boldsymbol{K}^f \otimes \boldsymbol{k}^*)\big)^T (\boldsymbol{K}^f \otimes \frac{\partial \boldsymbol{k}^*}{\partial \boldsymbol{u}_t}) \\ \left[\frac{\partial \boldsymbol{k}^*}{\partial \boldsymbol{u}_t}\right]_{ij} &= -k_i^* \boldsymbol{e}_j \boldsymbol{L}^{-1}(\boldsymbol{v}_t - \boldsymbol{v}_i)\end{aligned} \tag{16}$$

where $\boldsymbol{e}_j \in \mathbb{R}^{1\times(D+M)}$, $j = 1, 2, \cdots, D$ is a vector in which all components equal to 0, except the jth element to be 1. $k_i^* = k^v(\boldsymbol{v}_t, \boldsymbol{v}_i)$, $\boldsymbol{v}_i \in \boldsymbol{V}$, , $i = 1, 2, \cdots, N$ is covariance between the test point $\boldsymbol{v}_t$ and the ith training points in $\boldsymbol{V}$, $\boldsymbol{L}$ is the length scale in equation (2). According to equation (13)-(16), gradient-based methods can be utilized to find the solution of equation (10) such as the $fmincon$ function provided by the MATLAB optimization toolbox.

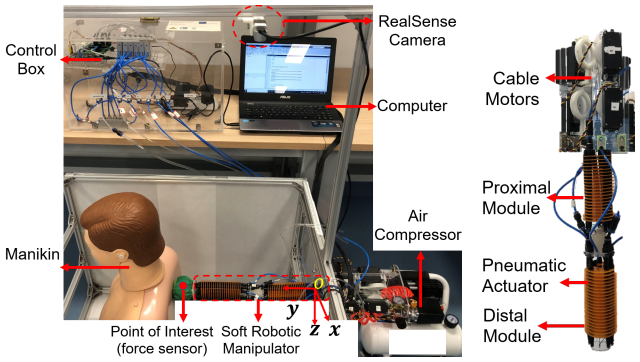
Procedures of the complete control policy is listed in Algorithm 2. To begin with, when a new interaction environment occurs, target-oriented proactive search is performed for T trials. Meanwhile, environment-specific data are collected to train the model parameters from the meta-learned initial parameters. After parameters learned, model-based optimal control actions drive the robot to desired performance. Lastly, the new environment data are recorded into the history dataset to prepare for the next new environment.

IV. Experiment Validation

This section presents an experimental validation of the proposed control approach, using a soft robotic manipulator system developed at Sant'Anna School of Advanced Studies by Manti and her co-authors [35].

A. Experimental Setup

The experimental setup (shown in Fig. 4a) consisted of: (1) a soft robotic manipulator: one of its end was installed with motors and mounted on an aluminum frame. The other end was covered by a green ball-like bath sponge, the center of which was the point of interest (PoI); (2) a manikin is used as the interaction environment for the soft robot; (3) pneumatic set-up: the control box contained six proportional pressure-controlled electronic valves (K8P Series, EVP Systems, Output: 0-3 bar), one filter (EVP Systems: MC-104FB0), one manometer (EVP Systems: M043-p12 0-12 bar), one Arduino Due. One standalone air compressor supplied pressure; (4) tendon set-up: six Hs-785hb Hitech Sail Winch Motors; (5) position sensor: a fixed camera (RealSense D435) detected the position of PoI that was marked by green color and the position of target that was marked by red color (as shown in Fig. 5); (6) force sensor: an ATI Nano25 force sensor was embedded in the bath ball to detect contact force; (7) the control algorithm was implemented in a computer (64-bit operating system, i5-6600 CPU, 3.30GHz). Since the success of wiping the manikin back mainly depended on the orthogonal force applied to the human body, so we controlled the tip position on the body surface (X and Z positions) and the tip force exerted in the Y direction (orthogonal force).

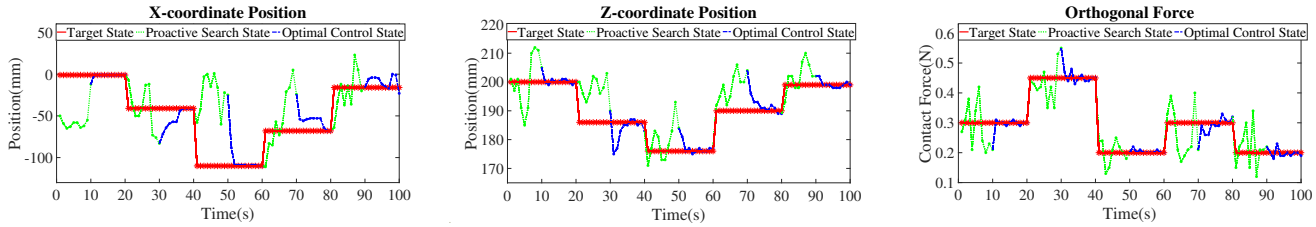


(a) (b)

Fig. 4: (a) Experimental setup. A manikin acts as the interaction environment for a soft robotic manipulator. One end of the robot is fixed as coordinate origin. Y-axis pointed to the manikin and Z-axis pointed to the ground. The other end is covered by a green bath ball and its center is point of interest (PoI). A force sensor is embedded in the bath ball to detect contact force. The air compressor supplies air pressure for pneumatic actuators. The control box contains proportional valves and Arduino boards. A RealSense camera detects the position of PoI. The control algorithm is implemented on a personal computer. (b) The structure of soft robotic manipulator consists of two identical modules and each module is a combination of pneumatic actuators and cables.

The soft robotic manipulator consists of two interconnected and identical modules as shown in Fig. 4b. Each module includes a combination of three pairs of McKibben-based actuators and three cables which are alternately displaced at an angle of 60° along a circle with a radius of 30mm. A layer-by-layer reinforcement structure is inserted along the entire module to constrain undesirable lateral and torsional movements. The total length of the soft robotic manipulator is 375mm and total weight is 220 grams. Cables and pneumatic actuators are decoupled, meaning that each one has a dedicated activation line for tension and pressure regulation, respectively. Meanwhile, the antagonistic actuation (pneumatic-cable) enables stiffness adaptation to tasks.

Fig. 5: Unknown changed interaction environments. The manikin was randomly moved such that both the interaction environment and the tracking target (i.e. the red label on the manikin's back) were changed. The soft robotic manipulator was controlled to be adaptive to the changing environment and track the target.



(a) (b) (c)

Fig. 6: Experimental results. Red solid lines are tracking targets in each new environment. Green dotted lines show that the robot takes proactive actions to search the new environment and thus the states appear more fluctuations. Blue dashed lines are optimal control states after model parameters have been adapted to the new environment. The actual states can finally reach the target states.

B. Experimental Results

Before experimentation, we applied random control actions to collect initial history data under three randomly selected environmental situations ($J = 3$ in Algorithm 1). Each situation contained 20 state-action pairs. During the experiments, the step size in equation (6) was set as 0.01 ($\alpha = 0.01$), the weight matrix in equation (7) was an identity matrix ($\boldsymbol{\Lambda} = \boldsymbol{I}$), the trial of target-oriented proactive search is 10 ($T = 10$ in Algorithm 2), the gradient step is 3 ($m = 3$ in Algorithm 1). To test the control performance of our method, we randomly moved the manikin's position to create unknown changing interaction environments, as shown in Fig. 5. The red label on the manikin's back was the tracking target that the soft robotic manipulator was controlled to achieve. The position of the red label was captured by the RealSense camera while the desired contact force was specified by the experimenter. The manikin was not moved until the current target was reached.

The experimental results are shown in Fig. 6 (see also supplemental video). Red solid lines are target states, green dotted lines are actual states during the proactive search stage, and blue dashed lines are actual states during the optimal control stage. The manikin was randomly moved 5 times corresponding to the tracking target changed 5 times. Each tracking target lasted 20 seconds during which 10 seconds were for proactive search and the remaining 10 seconds were for optimal control. We could see that when the soft robotic manipulator was in a previously unknown environment, the robot exploited its compliance advantage to search the new environment in a safe and proactive way. Actual states appeared fluctuations but within an accepted range. Since the data-driven model relies on environment-specific data to make state predictions, the proactive active search is performed in each new environment. After the proactive search stage and collected environment-specific data, the data-driven model was trained from the meta-learned initial parameters to adapt to the new environment. The actual states by optimal control showed more stable performance and could drive the robot to the target states.

V. Conclusion

In this work we developed a meta-learning-based optimal control approach for a soft robotic manipulator to interact with unknown environments. To begin with, the meta-learning exploits history data information to enable fast online model adaptation. In addition, the proactive search provides an efficient strategy for the robot to collect environment-specific data with minimal interaction trials. Lastly, model-based optimal control drives the robot to desired targets. Through our approach, a soft robotic manipulator could be controlled to achieve desired position and contact force simultaneously in unknown changing environments.

In our future work, we will compare our approach with other state-of-the-art methods such as the Neural-Fly method in [15] and the FAMLE method in [12]. On the other hand, this approach has limitations in situations where the interaction environment changes continuously. One possible solution is to develop an offline control policy structure combined with online adaptive hyper-parameters [14]. We will further improve the adaptation speed of our approach. Overall, this work applies meta-learning principles to the control of soft robots and demonstrates a viable control approach for the soft robotics community.

REFERENCES

[1] T. B. Sheridan, "Human–robot interaction: status and challenges," *Human factors*, vol. 58, no. 4, pp. 525–532, 2016.

[2] M. Vukobratovic, *Dynamics and robust control of robot-environment interaction*. World Scientific, 2009, vol. 2.

[3] X. Zhou, X. Wen, Z. Wang, Y. Gao, H. Li, Q. Wang, T. Yang, H. Lu, Y. Cao, C. Xu *et al.*, "Swarm of micro flying robots in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm5954, 2022.

[4] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[5] T. Belpaeme, J. Kennedy, A. Ramachandran, B. Scassellati, and F. Tanaka, "Social robots for education: A review," *Science robotics*, vol. 3, no. 21, p. eaat5954, 2018.

[6] R. Bemelmans, G. J. Gelderblom, P. Jonker, and L. De Witte, "Socially assistive robots in elderly care: a systematic review into effects and effectiveness," *Journal of the American Medical Directors Association*, vol. 13, no. 2, pp. 114–120, 2012.

[7] M. Benosman, "Model-based vs data-driven adaptive control: an overview," *International Journal of Adaptive Control and Signal Processing*, vol. 32, no. 5, pp. 753–776, 2018.

[8] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 1126–1135.

[9] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," *arXiv preprint arXiv:1803.02999*, 2018.

[10] G. Shi, K. Azizzadenesheli, M. O'Connell, S.-J. Chung, and Y. Yue, "Meta-adaptive nonlinear control: Theory and algorithms," *Advances in Neural Information Processing Systems*, vol. 34, pp. 10 013–10 025, 2021.

[11] A. Nagabandi, I. Clavera, S. Liu, R. S. Fearing, P. Abbeel, S. Levine, and C. Finn, "Learning to adapt in dynamic, real-world environments through meta-reinforcement learning," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=HyztsoC5Y7

[12] R. Kaushik, T. Anne, and J.-B. Mouret, "Fast online adaptation in robotics through meta-learning embeddings of simulated priors," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5269–5276.

[13] T. Anne, J. Wilkinson, and Z. Li, "Meta-learning for fast adaptive locomotion with uncertainties in environments and robot dynamics," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4568–4575.

[14] S. M. Richards, N. Azizan, J.-J. Slotine, and M. Pavone, "Adaptive-control-oriented meta-learning for nonlinear systems," *arXiv preprint arXiv:2103.04490*, 2021.

[15] M. O'Connell, G. Shi, X. Shi, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung, "Neural-fly enables rapid learning for agile flight in strong winds," *Science Robotics*, vol. 7, no. 66, p. eabm6597, 2022.

[16] T. Bock, C. Georgoulas, and T. Linner, "Towards robotic assisted hygienic services: Concept for assisting and automating daily activities in the bathroom," in *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, vol. 29. IAARC Publications, 2012, p. 1.

[17] Y. Ansari, M. Manti, E. Falotico, Y. Mollard, M. Cianchetti, and C. Laschi, "Towards the development of a soft manipulator as an assistive robot for personal care of elderly people," *International Journal of Advanced Robotic Systems*, vol. 14, no. 2, p. 1729881416687132, 2017.

[18] Z. Tang, P. Wang, W. Xin, and C. Laschi, "Learning-based approach for a soft assistive robotic arm to achieve simultaneous position and force control," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8315–8322, 2022.

[19] A. Bajo and N. Simaan, "Hybrid motion/force control of multi-backbone continuum robots," *The International journal of robotics research*, vol. 35, no. 4, pp. 422–434, 2016.

[20] H. Wang, H. Ni, J. Wang, and W. Chen, "Hybrid vision/force control of soft robot based on a deformation model," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 2, pp. 661–671, 2019.

[21] C. Della Santina, R. K. Katzschmann, A. Bicchi, and D. Rus, "Model-based dynamic feedback control of a planar soft robot: trajectory tracking and interaction with the environment," *The International Journal of Robotics Research*, vol. 39, no. 4, pp. 490–513, 2020.

[22] C. Della Santina, C. Duriez, and D. Rus, "Model based control of soft robots: A survey of the state of the art and open challenges," 2021. [Online]. Available: https://arxiv.org/abs/2110.01358

[23] T. George Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, "Control strategies for soft robotic manipulators: A survey," *Soft robotics*, vol. 5, no. 2, pp. 149–163, 2018.

[24] J. A. Farrell and T. Berger, "On the effects of the training sample density in passive learning control," in *Proceedings of 1995 American Control Conference-ACC'95*, vol. 1. IEEE, 1995, pp. 872–877.

[25] S. Konstantinov, A. Diveev, G. Balandina, and A. Baryshnikov, "Comparative research of random search algorithms and evolutionary algorithms for the optimal control problem of the mobile robot," *Procedia Computer Science*, vol. 150, pp. 462–470, 2019.

[26] A. T. Taylor, T. A. Berrueta, and T. D. Murphey, "Active learning in robotics: A review of control principles," *Mechatronics*, vol. 77, p. 102576, 2021.

[27] E. Yel, S. Gao, and N. Bezzo, "Meta-learning-based proactive online planning for uavs under degraded conditions," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 320–10 327, 2022.

[28] E. Brochu, V. M. Cora, and N. De Freitas, "A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," *arXiv preprint arXiv:1012.2599*, 2010.

[29] M. C. Yip and D. B. Camarillo, "Model-less hybrid position/force control: a minimalist approach for continuum manipulators in unknown, constrained environments," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 844–851, 2016.

[30] E. V. Bonilla, K. M. Chai, and C. Williams, "Multi-task gaussian process prediction," *Advances in neural information processing systems*, pp. 153–160, 2007.

[31] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer school on machine learning*. Springer, 2003, pp. 63–71.

[32] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in neural information processing systems*, vol. 25, 2012.

[33] J. Son, S. Gupta, and G. Tan, "Bayesian optimization in high dimensional input space," in *Proceedings of the 9th EAI International Conference on Simulation Tools and Techniques*, 2016, pp. 18–27.

[34] J. Quinonero-Candela, A. Girard, and C. E. Rasmussen, "Prediction at an uncertain input for gaussian processes and relevance vector machines-application to multiple-step ahead time-series forecasting," 2003.

[35] M. Manti, A. Pratesi, E. Falotico, M. Cianchetti, and C. Laschi, "Soft assistive robot for personal care of elderly people," in *2016 6th IEEE international conference on biomedical robotics and biomechatronics (BioRob)*. IEEE, 2016, pp. 833–838.