Learning task-parametrized assistive strategies for exoskeleton robots by multi-task reinforcement learning

Conference Paper · May 2017

DOI: 10.1109/ICRA.2017.7989695

CITATIONS
5

READS
232

5 authors, including:

Masashi Hamaya
OMRON SINIC X Corporation, Japan, Tokyo
11 PUBLICATIONS **93** CITATIONS

SEE PROFILE

Takamitsu Matsubara
Nara Institute of Science and Technology
102 PUBLICATIONS **1,028** CITATIONS

SEE PROFILE

Learning Task-Parametrized Assistive Strategies for Exoskeleton Robots by Multi-Task Reinforcement Learning

Masashi Hamaya[1,2], Takamitsu Matsubara[1,3], Tomoyuki Noda[1], Tatsuya Teramae[1] and Jun Morimoto[1]

***Abstract*— Recent studies suggest that reinforcement learning has great potential for generating assistive strategies in exoskeletons through physical interactions between a user and a robot. Previous methods focused on a task-specific assistive strategy, where for every single task (situation/context), the user needs to interact with a robot to learn an appropriate assistive strategy. Therefore, the learned strategies cannot be generalized for a new task. Since the sampling cost is expensive for such human-in-the-loop systems as exoskeletons, generalization must be enabled. In this paper, we propose to learn *task-parametrized assistive strategies* for exoskeleton robots. Our method employs an assistive strategy, which depends on the task parameter and the state variable, that can be learned from multiple sets of human-robot interaction data across different tasks and generalized even for an unseen task, given the task parameter without additional learning. To alleviate the user's burden in the learning process across multiple tasks, we exploit a data-efficient multi-task reinforcement learning framework. To verify the effectiveness of our method, we developed an experimental platform with an exoskeleton robot. We conducted a series of experiments whose experimental results show that our method can learn such a task-parametrized assistive strategy and be generalized for unseen tasks to reduce the user's electromyography signals (EMGs) during tasks.**

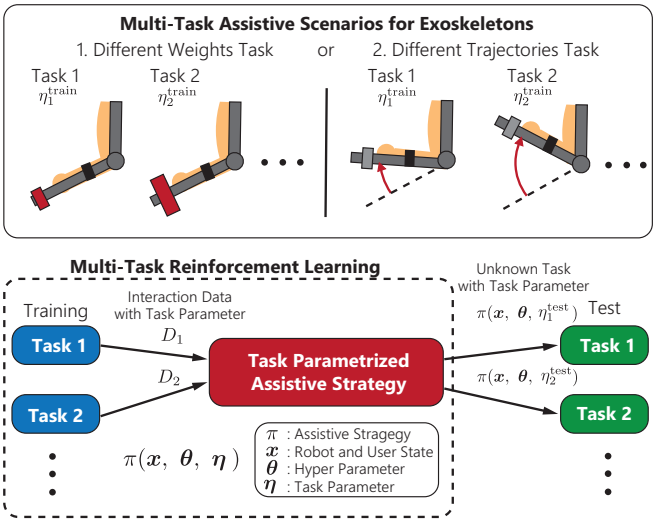


Fig. 1. Schematic diagram of our approach. We propose to learn a task-parametrized assistive strategy for exoskeleton robots. It can be learned with user-robot interaction data across multiple different training tasks with given task parameters, then, be generalized even for an unseen task given the task parameter without additional learning.

I. Introduction

Designing assistive strategies for exoskeleton robots and active orthoses has emerged as an important research topic because of many notable recent developments in robot hardware [1]–[8]. Nevertheless, this research topic remains challenging due to the complex and bidirectional physical interactions between robots and users. Several assistive strategies have already been explored. One typical strategy in power assistance, based on gravity compensation control [3], [4], [9], [10], effectively supports the load taken by users who are wearing an exoskeleton robot in a static posture. Another popular strategy is the electromyography (EMG)-based method [11], [12]. With the EMG-to-force model, the torque, which is required to compensate for the motion based on EMG signals, can be predicted. For walking and balancing assistance, an inverted pendulum model with Center of Mass (CoM) and Zero Moment Point (ZMP) can derive stable gait patterns [7], [13]. The adaptive oscillator-based strategy has also received much attention because of its simplicity. This approach uses an oscillator model to generate coordinated periodic trajectories with user intentions as references to a robot controller [2], [14]–[16]. A recent study combined this model with a state-machine-based controller [17].

As described above, most previous studies are based on mechanical models of human users, i.e., rigid-body dynamics or a CoM-ZMP inverted model, or only focus on periodic movements with oscillator models. Moreover, the interactions between users and robots are often not considered explicitly because of modeling difficulty. Therefore, such schemes suffer from modeling errors in the performance of the designed controller.

Recent studies also suggest that reinforcement learning has great potential for generating assistive strategies in exoskeletons through physical interactions between users and robots. Since it does not require a specific user model and does not suffer from modeling errors, good performance is expected for the learned strategy. One might be concerned about the heavy burden on users when interaction data are generated between a robot and a user. Our previous work, however,

[1]All authors are with the Department of Brain Robot Interface, ATR-CNS, Kyoto, Japan

[2]MH is with the Graduate School of Frontier Bioscience, Osaka University, Osaka, Japan

[3]TM is with the Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan

This study was supported by the Strategic Research Program for Brain Sciences from Japan Agency for Medical Research and development, AMED, by JSPS KAKENHI JP16H06565, by NEDO, by MIC-SCOPE, by "Development of Medical Devices and Systems for Advanced Medical Services" from AMED, by the ImPACT Program of Council for Science, Technology and Innovation (Cabinet Office, Government of Japan), and 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.

showed that a sample-efficient model-based reinforcement learning (RL) [18] effectively reduces the burden on users [19]. Huang et al. combined a model-based control scheme with RL to reduce the parameters that must be learned by RL and accelerated the learning speed [20], [21]. Other assistive strategies in different robots have also been explored with RL methods, including human-robot collaborative lifting [22], myoelectric prosthesis control [23], walking-aid robot control [24], robotic training for a dart-throwing task [25], and the telemanipulation of non-rigid objects [26].

However, all the previous methods focused on a task-specific assistive strategy, where for every single task (situation/context), the user needs to interact with the robot to learn an appropriate assistive strategy. Therefore, the learned strategies cannot be generalized for new tasks. Since the sampling cost is expensive for such human-in-the-loop systems as exoskeletons, generalization must be enabled. For example, to assist reaching movements by an exoskeleton, obtaining an assistive strategy with arbitrary targets is more desirable than with a single specific target.

In this paper, we propose to learn *task-parametrized assistive strategies* for exoskeleton robots. Our method employs an assistive strategy that depends on a task parameter and a state variable. Such a strategy can be learned from multiple sets of human-robot interaction data across different tasks and generalized even for an unseen task, given the task parameter without additional learning. To alleviate the user's burden in the learning process across multiple tasks, we exploit a data-efficient multi-task reinforcement learning framework [27].

To verify the effectiveness of our proposed method, we conducted a series of experiments and developed an experimental platform composed of a 1-DOF upper-limb exoskeleton robot for an elbow joint, EMG sensors, and a monitor. Then we implemented two different assistive scenarios: 1) trajectory tracking with different dynamics of different weights, and 2) different trajectories with several amplitudes (Fig. 1). Experimental results show that our method can learn such a task-parametrized assistive strategy and generalize it for unseen tasks to reduce the user's electromyography signals (EMGs) during the task.

The remainder of this paper is structured as follows. Section II presents our formulation of the learning problem of our assistive strategy. Section III shows our experimental study. Finally, we conclude this paper and describe future work in Section IV.

II. Learning Task-Parametrized Assistive Strategies by Reinforcement Learning

This section formulates the learning problem of assistive strategies that can be generalized across multiple tasks (Fig. 1). First, we summarize a learning framework for single-task assistive strategies by reinforcement learning [19] and extend it to formulate the learning problem of task-parametrized assistive strategies by multi-task reinforcement learning.

A. Single-Task Formulation

We assume that the robot is tightly coupled to the user. The future state of the robot and the user depends on the current state of the robot, its action, and the user state and action. Therefore, human-robot integrated dynamics can be represented as follows [19]:

$$\boldsymbol{x}_{t+1} = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t) + \boldsymbol{\xi}_t, \quad \boldsymbol{\xi}_t \sim \mathcal{N}(0, \boldsymbol{\Sigma}_\xi), \tag{1}$$

where

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{s} \\ \boldsymbol{v} \end{bmatrix}, \quad \Sigma_\xi = \begin{bmatrix} \boldsymbol{\Sigma}_\epsilon & 0 \\ 0 & \boldsymbol{\Sigma}_\zeta \end{bmatrix}, \tag{2}$$

where $\boldsymbol{s}_t$ is the robot state (e.g., joint angles and velocities) and $\boldsymbol{u}_t$ is the robot action (e.g., joint torques or pressures generated by the actuators). $\boldsymbol{v}_t$ is the user's action (e.g., muscle activations), and $\boldsymbol{\epsilon}_t$ and $\boldsymbol{\zeta}_t$ are additive Gaussian noises.

Based on the above system, we formulate our learning problem of assistive strategies as a policy search problem. Our objective is to learn robot control policy (assistive strategy) π: $\pi(\boldsymbol{x}, \boldsymbol{\theta}) = \boldsymbol{u}$ from user-robot interaction data that can minimize the long-term cost:

$$J^\pi(\boldsymbol{\theta}) = \sum_{t=0}^{T} \mathbb{E}_{\boldsymbol{x}_t}[c(\boldsymbol{x}_t)], \ \boldsymbol{x}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) \tag{3}$$

where J^π evaluates the cost for T steps, $\boldsymbol{\theta}$ is a policy parameter, and $c(\boldsymbol{x}_t)$ is the cost in state $\boldsymbol{x}$ at time t.

Note that unlike typical autonomous robot control problems, cost function $c(\boldsymbol{x}_t)$ does not incorporate such task dependent information as the desired target locations or trajectories. Instead, our cost function only considers the user's muscular effort that can be measured by EMGs because the robot needs to *assist* the user's voluntary motions rather than control them. Such a cost function design, which allows the user to lead the learning process, the can result in an appropriate assistive strategy [19].

To achieve such efficient learning, our previous work used a data-efficient model-based reinforcement learning framework called PILCO [18], which learns a probabilistic dynamics model to achieve a sample-efficient policy search rather than directly learning it from data. More details of the learning process are shown below.

1) Model Learning: PILCO learns a dynamical model using Gaussian process regression [28], where $(\boldsymbol{x}_t, \boldsymbol{u}_t) \in \mathbb{R}^{D+F}$ is the training input and $\boldsymbol{\Delta}_t = \boldsymbol{x}_{t+1} - \boldsymbol{x}_t \in \mathbb{R}^D$ is the training output. It typically uses the following kernel function:

$$k(\tilde{\boldsymbol{x}}_p, \tilde{\boldsymbol{x}}_q) = {\sigma_f}^2 \exp\left(-\frac{1}{2}(\tilde{\boldsymbol{x}}_p - \tilde{\boldsymbol{x}}_q)^\top \boldsymbol{\Lambda}^{-1}(\tilde{\boldsymbol{x}}_p - \tilde{\boldsymbol{x}}_q)\right) + \delta_{pq}\sigma_\xi, \tag{4}$$

where $\tilde{\boldsymbol{x}} := [\boldsymbol{x}^\top, \boldsymbol{u}^\top]$, $\boldsymbol{\Lambda}$ is a diagonal matrix that expresses the characteristic length, σ_f is the bandwidth, and σ_{ξ_m} is a noise parameter. These parameters are learned with n training inputs $\tilde{\boldsymbol{X}} = [\tilde{\boldsymbol{x}}_1, ..., \tilde{\boldsymbol{x}}_n]$ and targets $\boldsymbol{y} = [\boldsymbol{\Delta}_1, ..., \boldsymbol{\Delta}_n]$.

The predictive distribution of $\boldsymbol{x}_{t+1}$ is analytically given as follows:

$$p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_t) = \mathcal{N}(\boldsymbol{x}_{t+1}|\boldsymbol{\mu}_{t+1}, \boldsymbol{\Sigma}_{t+1}), \tag{5}$$

$$\boldsymbol{\mu}_{t+1} = \boldsymbol{x}_t + \mathbb{E}_f[\boldsymbol{\Delta}_t], \quad \boldsymbol{\Sigma}_{t+1} = \mathrm{Var}_f[\boldsymbol{\Delta}_t], \tag{6}$$

where

$$\mathbb{E}_f[\boldsymbol{\Delta}_t] = \boldsymbol{k}_*{}^\top(\boldsymbol{K} + {\sigma_\xi}^2\boldsymbol{I})^{-1}\boldsymbol{y} \tag{7}$$

$$\mathrm{Var}_f[\boldsymbol{\Delta}_t] = k_{**} - \boldsymbol{k}_*{}^\top(\boldsymbol{K} + \sigma_\xi^2\boldsymbol{I})^{-1}\boldsymbol{k}_* \tag{8}$$

$\boldsymbol{k}_* := k(\tilde{\boldsymbol{X}}, \tilde{\boldsymbol{x}}_t)$, where $k_{**} := k(\tilde{\boldsymbol{x}}_t)$, and $\boldsymbol{K}$ is a kernel matrix, each of which element follows $K_{ij} = k(\tilde{x}_i, \tilde{x}_j)$.

2) Control Policy: We employed the following preliminary control policy:

$$\tilde{\pi}(\boldsymbol{x}_*) = \sum_{i=1}^{N} k(\boldsymbol{g}_i, \boldsymbol{x}_*)(\boldsymbol{K} + \sigma_\pi^2\boldsymbol{I})^{-1}\boldsymbol{t} = k(\boldsymbol{G}, \boldsymbol{x}_*)^\top\boldsymbol{\alpha} \tag{9}$$

$\boldsymbol{\alpha} = (\boldsymbol{K} + \sigma_\pi^2\boldsymbol{I})^{-1}\boldsymbol{t}$, where $\boldsymbol{x}_*$ is test input, $\boldsymbol{t}$ is a training target, $\boldsymbol{G} = [\boldsymbol{g}_1, ..., \boldsymbol{g}_N]$ are the centers of the Gaussian basis functions, σ_π^2 is noise variance, and k is a kernel function. Base policy $\bar{\pi}$ is designed as

$$\bar{\pi}(\boldsymbol{x}_*) = \boldsymbol{u}_{\max}\sigma(\tilde{\pi}(\boldsymbol{x}_*)). \tag{10}$$

For the control limit, we utilize $\boldsymbol{u}_{\max}$ and squashing function $\sigma(x) \in [-1, 1]$.

3) Policy Evaluation: To evaluate the control policy, we need to compute long-term cost J^π. Although it cannot be obtained analytically due to the Gaussian process model's complexity, PILCO employs a reasonable approximation scheme with an analytic moment-matching technique.

To predict $\boldsymbol{x}_{t+1}$, PILCO assumes distribution $p(\tilde{\boldsymbol{x}}_t) = p(\boldsymbol{x}_t, \boldsymbol{u}_t)$ is a Gaussian distribution and calculates $p(\boldsymbol{\Delta}_t)$ as follows:

$$p(\boldsymbol{\Delta}_t) = \iint p(f(\tilde{\boldsymbol{x}}_t)|\tilde{\boldsymbol{x}}_t)p(\tilde{\boldsymbol{x}}_t)\mathrm{d}f\mathrm{d}\tilde{\boldsymbol{x}}_t. \tag{11}$$

Based on this predictive distribution, expected cost $\mathbb{E}_{\boldsymbol{x}_t}[c(\boldsymbol{x}_t)]$ can be computed analytically:

$$\mathbb{E}_{\boldsymbol{x}_t}[c(\boldsymbol{x}_t)] = \int c(\boldsymbol{x}_t)p(\boldsymbol{x}_t)\mathrm{d}\boldsymbol{x}_t. \tag{12}$$

With the above equations, we can analytically compute the approximation of J^π.

4) Policy Improvement with Analytic Gradient: A policy improvement is to find $\boldsymbol{\theta}$, which minimizes $J^\pi(\boldsymbol{\theta})$. $\boldsymbol{\theta}$ is composed of the center of Gaussian basis functions $\boldsymbol{G}$, the length scale of the kernel functions, and targets $\boldsymbol{t}$ in Eq. (9). Gradient $\partial J^\pi(\boldsymbol{\theta})/\partial\boldsymbol{\theta}$ can be computed analytically using the chain-rule because of the policy evaluation's analytic expression. The gradient is expressed by $\varepsilon_t := \mathbb{E}_{\boldsymbol{x}_t}[c(\boldsymbol{x}_t)]$

$$\begin{aligned}\frac{\mathrm{d}J^\pi(\boldsymbol{\theta})}{\mathrm{d}\boldsymbol{\theta}} &= \sum_{t=1}^{T}\frac{\mathrm{d}\varepsilon_t}{\mathrm{d}\boldsymbol{\theta}},\\ \frac{\mathrm{d}\varepsilon_t}{\mathrm{d}\boldsymbol{\theta}} &= \frac{\mathrm{d}\varepsilon_t}{\mathrm{d}p(\boldsymbol{x}_t)}\frac{\mathrm{d}p(\boldsymbol{x}_t)}{\mathrm{d}\boldsymbol{\theta}} := \frac{\partial\varepsilon_t}{\mathrm{d}\boldsymbol{\mu}_t}\frac{\mathrm{d}\boldsymbol{\mu}_t}{\partial\boldsymbol{\theta}} + \frac{\partial\varepsilon_t}{\mathrm{d}\boldsymbol{\Sigma}_t}\frac{\mathrm{d}\boldsymbol{\Sigma}_t}{\partial\boldsymbol{\theta}}.\end{aligned} \tag{13}$$

Therefore, standard gradient-based non-convex optimization methods, such CG or BFGS, can be applied to find a locally optimal parameter $\boldsymbol{\theta}$.

By repeatedly applying the above model learning and policy improvement, we can efficiently learn an assistive strategy just from the user-robot interaction data [19].

B. Multi-Task Formulation

One practical issue of the above single-task approach is that for every single task (situation/context), the user needs to interact with the robot to learn an appropriate assistive strategy: the lack of generalization capability. To allow generalization, we consider a task-parametrized assistive strategy $\pi : \pi(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\eta}) = \boldsymbol{u}$, a function of state $\boldsymbol{x}$, policy parameter $\boldsymbol{\theta}$, and task parameter $\boldsymbol{\eta}$. We assume that the task parameter is given for each task and containing task specific information. For different tasks, different actions may be required even at the same state $\boldsymbol{x}$. With this strategy, the state $\boldsymbol{x}$ and the policy parameter $\boldsymbol{\theta}$, we hope to generate different actions suitable for different tasks by only changing the task parameter η.

To find such a task-parametrized assistive strategy suitable to generalize across multiple tasks, we can use a data-efficient multi-task RL scheme based on PILCO (multi-task PILCO) [27]. This scheme aims to learn a policy parameter $\boldsymbol{\theta}$ given multiple training tasks with corresponding task parameters η^{train}, that can also be effective for unseen test tasks with task parameters η^{test}. Such a learning problem is formulated as to find an optimal policy parameter $\boldsymbol{\theta}^*$ in the task-parametrized assistive strategy π which minimizes the long-term *average* cost function across M training tasks with the task parameters $\eta_{1:M}^{\mathrm{train}} = \{\eta_1, \cdots, \eta_M\}$:

$$J^\pi(\boldsymbol{\theta}, \eta_{1:M}^{\mathrm{train}}) = \frac{1}{M}\sum_{m=0}^{M} J_m^\pi(\boldsymbol{\theta},\ \eta_m^{\mathrm{train}}). \tag{14}$$

While the generalization capability are implicitly determined by the specific choices of the function π and the task parameter η, its effectiveness was demonstrated for several real robot experiments [27].

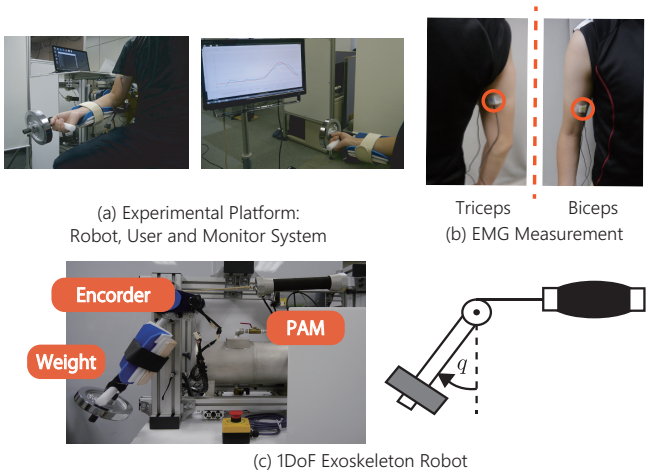
For learning assistive strategies in exoskeletons, we made two modifications in the original multi-task PILCO. We assume that the dynamics are task-dependent f_m because in Eq. (1), the user's response depends on the task (e.g., the desired trajectory or goal targets), unlike typical robot control applications. On the other hand, the cost function can be common across multiple tasks because it only depends on the user's muscular effort, as explained in Section II.A.

In this paper, we focus on the following task-parametrized policy:

$$\pi(\boldsymbol{x}, \boldsymbol{\theta}, \eta) = \eta\bar{\pi}(\boldsymbol{x}, \boldsymbol{\theta}), \tag{15}$$

where the basis policy in Eq. (9) is multiplied by the task parameter to simplify our framework, and to expect its high generalization capability. We can derive the policy improvement scheme:

$$\frac{\partial J^\pi}{\partial\boldsymbol{\theta}} = \sum_{m=1}^{M}\frac{\partial J_m^\pi}{\partial\boldsymbol{\theta}}, \tag{16}$$



(a) Experimental Platform: Robot, User and Monitor System

(b) EMG Measurement

(c) 1DoF Exoskeleton Robot

Fig. 2. Experimental setup. (a) shows the experimental platform composed of the robot, user and monitor. (b) shows the EMGs sensors placed on biceps and triceps. (c) shows the 1-DOF upper limb exoskeleton robot.

where

$$J_m^{\pi}(\boldsymbol{\theta}) = \sum_{t=0}^{T} \int c(\boldsymbol{x}_t) p_m(\boldsymbol{x}_t) \mathrm{d}\boldsymbol{x}_t \tag{17}$$

and $p_m(\boldsymbol{x})$ is the state distribution based on dynamics f_m and assistive strategy $\pi(\boldsymbol{x}, \boldsymbol{\theta}, \eta_m^{\text{train}})$. The gradient of each task $\partial J_m^{\pi} / \partial \boldsymbol{\theta}$ can be computed identically as a single task because of the summation of the long-term cost and multiplying the parameter on all tasks, as shown in Eqs. (11)-(13).

III. Experiment

To verify our approach, we conducted a series of subjective experiments. We developed an experimental platform that included a subject, a 1-DOF upper-limb elbow-joint exoskeleton robot, EMG sensors, and a monitor (Fig. 2 (a)). A reaching task was selected for the evaluation. The subject watched the monitor and tried to move the robot. We considered two different scenarios: 1) trajectory tracking with different robot dynamics and different weights and 2) trajectory tracking with different amplitudes of the trajectories. We investigated whether our approach can learn suitable task-parametrized assistive strategies by multi-task reinforcement learning and whether the learned strategies can be generalized across multiple unseen tasks without additional learning.

We commonly set the cost function for both scenarios:

$$\begin{aligned} c(\boldsymbol{x}_t) &= 1 - \exp\left(-\frac{1}{2\sigma_c^2} \boldsymbol{x}_t^{\top} \boldsymbol{T}^{-1} \boldsymbol{x}_t \right) \\ \boldsymbol{T} &= \begin{pmatrix} 0 & 0 \\ 0 & \boldsymbol{T}_v, \end{pmatrix} \end{aligned} \tag{18}$$

where σ_c is the width of the cost function and $\boldsymbol{T}$ is a diagonal matrix that expresses the weight of each element for the cost state.

A. Experimental Setup

We placed 2 ch of EMGs on the biceps and the triceps (Fig. 2(b)) to measure the user's physical effort. The sampling time was 0.004 s. They were rectified and low-pass filtered, the cutoff frequency was 2.0 Hz, 2nd order Butterworth filter. The robot was driven by a pneumatic artificial muscle (PAM) actuator (FESTO), the link length was 0.4 m, and it weighed 1.7 kg (Fig. 2(c)). We set the pressure limitation of the actuator, 0.8 MPa for safe user-robot interactions.

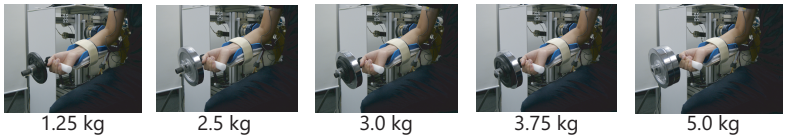
We utilized a PILCO open source code [29] and modified it for multi-task learning. The subject and robot states in Eq. (1) were $\boldsymbol{x} = [q \ \dot{q} \ E_b \ E_t]^{\top}$, where q was the robot's joint angle, $\dot{q}$ was the angular velocity, and E_b and E_t were the filtered-biceps and -triceps EMGs that were averaged for 0.2 s. u was the input pressure of the PAM's MPa. The control period of assistive strategy dt was 0.2s, and the prediction horizon was 2.0 s. The weight of the cost function was $T_v = \mathrm{diag}([2.0 \ 4.0])$ and the cost width σ_c was 0.5 in Eq. (3). The number of basis functions N in Eq. (9) was 100. One multi-/single-task learning session was composed of five initial trials for the data collections and five learning trials on each task.

1) Different Weights Scenarios: The subject performed two training tasks with 2.5 and 3.75 kg weights. The task parameter, which was given as $\boldsymbol{\eta}^{\text{train}} = [1.0 \ 1.5]$, was determined by the weight ratio against 2.5 kg. Then we applied the learned assistive strategy to the three test tasks with 1.25, 3.0, and 5.0 kg weights. The test task parameter was $\boldsymbol{\eta}^{\text{test}} = [0.5 \ 1.2 \ 2.0]$. $u_{\max}$, which was used in Eq. (10), was 0.3 MPa. The trajectory was a bell-shaped curve. The random pressure for the initial interaction was given as $u_{\text{init}} \sim \mathcal{N}(\eta^{\text{train}} A \sin(0.5\pi dt), \ 0.2)$, where A was 0.3 MPa amplitude of pressure. For comparisons, the subject performed single-task learning with 2.5 or 3.75 kg weights. We also tested the learned strategy with the other weights. In the single-task learning, we didn't use the task parameter except for the random initial interaction. The test and training conditions for the methods are summarized in Fig. 3.

2) Different Trajectories Scenarios: The subject performed two training tasks. The trajectories were bell-shaped curves, and the amplitudes were $2\pi/3$ and $5\pi/6$ rad. The task parameter was $\boldsymbol{\eta}^{\text{train}} = [0.8 \ 1.0]$ and was determined by the ratio of the amplitude against $5\pi/6$. Then we applied the learned assistive strategy to the three test tasks with $7\pi/12$, $3\pi/4$, and $11\pi/12$ rad amplitudes. The test task parameter was $\boldsymbol{\eta}^{\text{test}} = [0.7 \ 0.9 \ 1.1]$. $u_{\max}$ was 0.2 MPa. The random pressure was given as $u_{\text{init}} \sim \mathcal{N}(\eta^{\text{train}} A \sin(0.5\pi dt), \ 0.2)$, where $A = 0.2$ MPa. For comparisons, the subject performed a single training task with $2\pi/3$ or $5\pi/6$ rad amplitudes. The test and training conditions are summarized in Fig. 4.

B. Experimental Results

The robot learned the assistive strategies for the session composed of five initial and five learning trials corresponding to a total of 40s of data on the multi-task, 20s on the single-task. The subject conducted the session three times.



Task \ Weight kg	1.25	2.5	3.0	3.75	5.0
Multi-Task(2.5, 3.75 kg)	Test	Train	Test	Train	Test
Single-Task(2.5 kg)	Test	Train	Test	Test	Test
Single-Task(3.75 kg)	Test	Test	Test	Train	Test

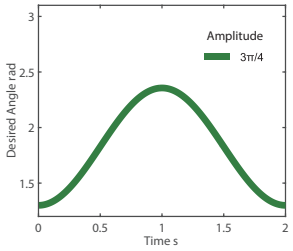
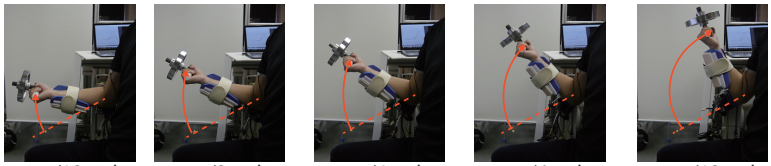


Fig. 3. Experimental setup for the different weights task. The subject performed the trajectory tracking tasks. The pictures show the weights on the each tasks. The right figure shows the reference trajectory. The table shows the setting of the training and test trials on the multi or single-tasks.



Task \ Amplitude rad	7π/12	2π/3	3π/4	5π/6	11π/12
Multi-Task(2π/3, 5π/6 rad)	Test	Train	Test	Train	Test
Single-Task(2π/3 rad)	Test	Train	Test	Test	Test
Single-Task(5π/6 rad)	Test	Test	Test	Train	Test

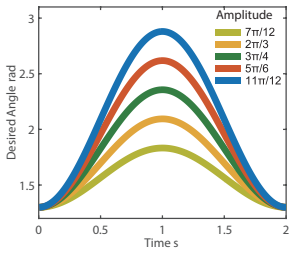


Fig. 4. Experimental setup for the different trajectories task. The pictures and figure show the reference trajectories on the each tasks. The table shows the setting of the training and test trials on the multi or single-tasks.

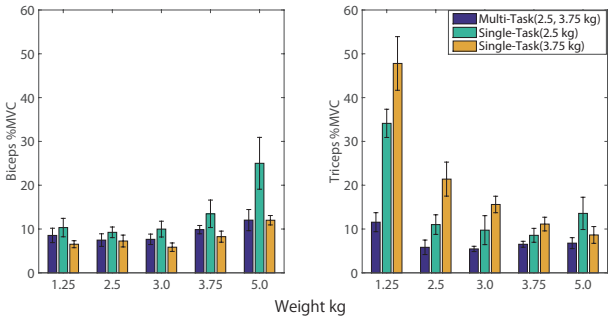


Fig. 5. Experimental results of the mean %MVC on the different weights scenario. For the most of the case, the multi-task (blue bar) shows the smallest %MVC. The single task biceps %MVC (2.5 kg, green bar) increased with the large weight and single-task (3.75 kg, orange bar) triceps %MVC increased with the light weights.

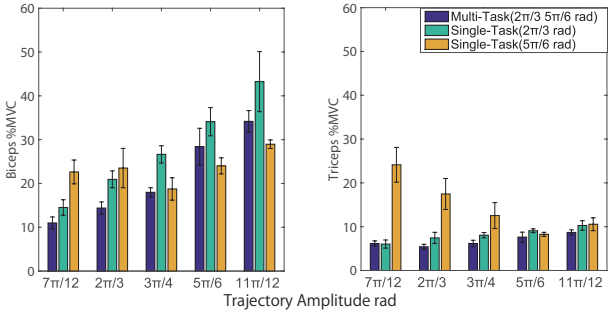


Fig. 6. Experimental results of %MVC on the different trajectories scenario. The results were consistent to the different weights scenario.

Among the three sessions, the learned strategy, which was the minimum accumulated cost, was utilized in the test. The subject tried the test five times with this strategy.

1) Different Weights Scenarios: Figure 5 shows the mean EMGs of each task. The EMGs were normalized by the percentages of maximum voluntary contraction (%MVC): %MVC = $E/E_{\max}$, where $E_{\max}$ was the maximum value of the EMGs. The blue bar shows the EMGs with the multi-task strategy, a green bar with a single-task (2.5 kg), and an orange bar with a single-task (3.75 kg). For most of the cases, multi-task learning with the task-parametrized strategy resulted in the smallest EMGs for the biceps and the triceps, even for unseen tasks as 1.25, 3, and 5 kg among all three methods. On the other hand, for the single-task strategy trained with a light weight (2.5 kg), the EMGs greatly increased as the weight was increased in the biceps muscle and vice versa for the strategy trained with a heavy weight (3.75 kg) in the triceps muscle. The maximum tracking errors against the amplitude of all three methods was only 7.5%, sufficiently small and comparative.

2) Different Trajectories Scenarios: Figure 6 shows the mean EMGs on each task. The blue bar shows the EMGs with the multi-task strategy, a green bar with a single-task ($2\pi/3$ rad), and an orange bar with a single-task ($5\pi/6$ rad). The results are consistent with the case of different weight scenarios. Multi-task learning with the task-parametrized strategy resulted in the smallest EMGs for the biceps and the triceps, even for unseen tasks among all three methods. For the single-task strategy trained with a small amplitude of trajectory ($2\pi/3$ rad), the EMGs greatly increased as the amplitude increased in the biceps and vice versa for the

strategy trained with a large weight ($5\pi/6$ rad) in the triceps muscle. The maximum tracking errors against the amplitude of all three methods was only 7.8%, sufficiently small and comparative.

These experimental results verified the effectiveness of our method for learning a task-parametrized assistive strategy by a multi-task RL and confirmed its generalization capability even for unseen tasks without additional learning.

IV. Conclusion

In this paper, we proposed to learn task-parametrized assistive strategies for exoskeleton robots. Our method employed an assistive strategy that depends on the task parameter and the state variable. It can be learned from multiple data sets across different tasks and generalized even for an unseen task given a task parameter without additional learning. To alleviate the user's burden in the learning process, we exploited a data-efficient multi-task RL. To verify the effectiveness of our proposed method, we conducted series of experiments whose results show that it can learn such a task-parametrized assistive strategy and be generalized for unseen tasks to reduce the user's EMGs.

As for future works, we will apply our method to a lower exoskeleton robot with multiple DOFs [5], [30] for squatting assistance with multiple weights and targets, and we will extend our framework for multi-user scenarios to generalize the learned strategies from one user to another [31].

References

[1] K. Suzuki, G. Mito, H. Kawamoto, Y. Hasegawa, and Y. Sankai, "Intention-based walking support for paraplegia patient with robot suithal," *Advanced Robotics*, vol. 21, no. 12, pp. 1441–1469, 2007.

[2] X. Zhang and M. Hashimoto, "SBC for motion assist using neural oscillator," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, 2009, pp. 659–664.

[3] S. Jacobsen, "On the development of XOS, a powerful exoskeletal robot," in *Plenary Talk IEEE/RSJ Int'l Conf. on Intelligent Robots and System*, 2007.

[4] K. Kazerooni, A. Chu, and R. Steger, "That which does not stabilize, will only make us stronger," *The Int'l J. of Robotics Research*, vol. 26, no. 1, pp. 75–89, 2007.

[5] S. Hyon, J. Morimoto, T. Matsubara, T. Noda, and M. Kawato, "XoR: Hybrid Drive Exoskeleton Robot That Can Balance," in *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, 2011, pp. 2715–2722.

[6] A. Asbeck, S. De Rossi, I. Galiana, Y. Ding, and C. Walsh, "Stronger, smarter, softer: Next-generation wearable robots," *IEEE Robotics Automation Magazine*, vol. 21, no. 4, pp. 22–33, Dec 2014.

[7] S. Wang, L. Wang, C. Meijneke, E. van Asseldonk, T. Hoellinger, G. Cheron, Y. Ivanenko, V. La Scaleia, F. Sylos-Labini, M. Molinari, F. Tamburella, I. Pisotta, F. Thorsteinsson, M. Ilzkovitz, J. Gancet, Y. Nevatia, R. Hauffe, F. Zanow, and H. van der Kooij, "Design and control of the mindwalker exoskeleton," *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 2, pp. 277–286, March 2015.

[8] R. Farris, H. Quintero, and M. Goldfarb, "Preliminary evaluation of a powered lower limb orthosis to aid walking in paraplegic individuals," *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, vol. 19, no. 6, pp. 652–659, Dec 2011.

[9] S. Banala, S. Agrawal, A. Fattah, V. Krishnamoorthy, W.-L. Hsu, J. Scholz, and K. Rudolph, "Gravity-balancing leg orthosis and its performance evaluation," *IEEE Trans. on Robotics*, vol. 22, no. 6, pp. 1228–1239, 2006.

[10] C. J. Walsh, K. Endo, and H. Herr, "A quasi-passive leg exoskeleton for load-carrying augmentation," *Int'l J. of Humanoid Robotics*, vol. 4, no. 3, pp. 487–506, 2007.

[11] H. Kawamoto, S. Kanbe, and Y. Sankai, "Power assist method for HAL-3 using EMG-based feedback controller," in *Proc. Int'l Conf. on Systems, Man and Cybernetics*, 2003, pp. 1648–1653.

[12] C. Fleischer, C. Reinicke, and G. Hommel, "Predicting the intended motion with EMG signals for an exoskeleton orthosis controller," in *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, 2005, pp. 2029–2034.

[13] T. Kagawa and Y. Uno, "Gait pattern generation for a power-assist device of paraplegic gait," in *Proc. IEEE Int'l Symposium on Robot and Human Interactive Communication*, 2009, pp. 633–638.

[14] R. Ronsse, N. Vitiello, T. Lenzi, J. van den Kieboom, M. Carrozza, and A. Ijspeert, "Adaptive oscillators with human-in-the-loop: Proof of concept for assistance and rehabilitation," in *Proc. IEEE/RAS-EMBS Int'l Conf. on Biomedical Robotics and Biomechatronics*, 2010, pp. 668–674.

[15] T. Matsubara, A. Uchikata, and J. Morimoto, "Full-body exoskeleton robot control for walking assistance by style-phase adaptive pattern generation," in *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, 2012, pp. 3914–3920.

[16] T. Matsubara, D. Uto, T. Noda, T. Teramae, and J. Morimoto, "Style-phase adaptation of human and humanoid biped walking patterns in real systems," in *Proc. IEEE/RAS Int'l Conf. on Humanoid Robots*, 2014, pp. 128–133.

[17] T. Yan, M. Cempini, C. M. Oddo, and N. Vitiello, "Review of assistive strategies in powered lower-limb orthoses and exoskeletons," *Robotics and Autonomous Systems*, vol. 64, pp. 120 – 136, 2015.

[18] M. P. Deisenroth, D. Fox, and C. E. Rasmussen, "Gaussian processes for data-efficient learning in robotics and control," *IEEE Trans. on Pattern Recognition and Machine Intelligence*, vol. 37, no. 2, pp. 408–423, 2015.

[19] M. Hamaya, T. Matsubara, T. Noda, T. Teramae, and J. Morimoto, "Learning assistive strategies from a few user-robot interactions: Model-based reinforcement learning approach," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, 2016, pp. 3346–3351.

[20] R. Huang, H. Cheng, H. Guo, Q. Chen, H.-T. Tran, and X. Lin, "Interactive learning for sensitivity factors of a human-powered augmentation lower exoskeleton," in *Proc. IEEE/RAS Int'l Conf. on Intelligent Robots and Systems*, 2015, pp. 6409–6415.

[21] R. Huang, H. Cheng, H. Guo, Q. Chen, and X. Lin, "Hierarchical interactive learning for a human-powered augmentation lower exoskeleton," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, 2016, pp. 257–263.

[22] T. Tamei and T. Shibata, "Fast Reinforcement Learning for Three-Dimensional Kinetic Human-Robot Cooperation with an EMG-to-Activation Model," *Advanced Robotics*, vol. 25, no. 5, pp. 563–580, 2011.

[23] P. Pilarski, M. Dawson, T. Degris, F. Fahimi, J. Carey, and R. Sutton, "Online human training of a myoelectric prosthesis controller via actor-critic reinforcement learning," in *Proc. IEEE/RAS-EMBS Int'l Conf. on Rehabilitation Robotics*, 2011, pp. 1–7.

[24] W. Xu, J. Huang, Y. Wang, and H. Cai, "Study of reinforcement learning based shared control of walking-aid robot," in *Proc. IEEE/SICE Int'l Symposium on System Integration*, 2013, pp. 282–287.

[25] C. Obayashi, T. Tamei, and T. Shibata, "Assist-as-needed robotic trainer based on reinforcement learning and its application to dart-throwing," *Neural Networks*, vol. 53, pp. 52–60, 2014.

[26] T. Matsubara, T. Hasegawa, and K. Sugimoto, "Reinforcement Learning of Shared Control for Dexterous Telemanipulation: Application to a Page Turning Skill," in *Proc. IEEE Int'l Symposium on Robot and Human Interactive Communication*, 2015, pp. 343–348.

[27] M. P. Deisenroth, P. Englert, J. Peters, and D. Fox, "Multi-task policy search for robotics," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, 2014, pp. 3876–3881.

[28] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. Springer, 2006.

[29] "Pilco web site." [Online]. Available: http://mlg.eng.cam.ac.uk/pilco/

[30] B. Ugurlu, C. Doppmann, M. Hamaya, P. Forni, T. Teramae, T. Noda, and J. Morimoto, "Variable ankle stiffness improves balance control: Experiments on a bipedal exoskeleton," *IEEE Trans. on Mechatronics*, vol. 21, no. 1, pp. 79–87, 2016.

[31] T. Matsubara and J. Morimoto, "Bilinear modeling of emg signals to extract user-independent features for multiuser myoelectric interface," *IEEE Trans. on Biomedical Engineering*, vol. 60, no. 8, pp. 2205–2213, 2013.