AutoEncoder-based Safe Reinforcement Learning for Power Augmentation in a Lower-limb Exoskeleton

2021 9th RSI International Conference on Robotics and Mechatronics (ICRoM) | 978-1-6654-2094-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICRoM54204.2021.9663471

Mohammadreza Abbasi
Department of Mechanical Engineering
Sharif University of Technology
Tehran, Iran
mr.abbasi.sharif@gmail.com

Mohammad Karami
Department of Mechanical Engineering
Sharif University of Technology
Tehran, Iran
karami.mohammad@mech.sharif.edu

Amirreza Koushki
Department of Mechanical Engineering
Sharif University of Technology
Tehran, Iran
koushki1376@gmail.com

Gholamreza Vossoughi
Department of Mechanical Engineering
Sharif University of Technology
Tehran, Iran
vossough@sharif.edu

***Abstract*— Power augmentation in wearable robots has recently attracted the attention of researchers in industry and academia. Human-robot synchronized control is one of the most substantial goals that needs to be achieved. The change of gait patterns between different users and cycles, and the close interaction of human and robot necessitates the estimation of human motion intention in real-time. Reinforcement learning (RL) is beneficial in such situations mainly due to its online and model-free nature. However, RL algorithms struggle in high-dimensional continuous search spaces in terms of convergence speed and safety. Therefore, this paper proposes an RL-based control strategy for power augmentation in a lower-limb exoskeleton to extract the human motion and minimize the interaction force. A deep AutoEncoder (AE) neural network is trained and employed for dimensionality reduction to improve the convergence speed and safety. Simulation and two experiments were carried out with a lower-limb exoskeleton on a healthy participant. The results indicate the efficacy of the proposed control strategy and its superior performance compared to model-based approaches while improving speed and safety.**

Keywords—Reinforcement Learning, Deep Autoencoder, Power Augmentation, Interaction Force, Lower-limb Exoskeleton

I. INTRODUCTION

Exoskeletons are electromechanical devices that have lately gotten a great interest in academic and industrial settings as a power augmentation technology. Coordinated control of the exoskeleton and Human-Robot movement synchronization are two major challenges addressed by researchers in such applications. This is due to the close interaction between the robot and the user, as well as the fact that the movement pattern varies considerably across individuals and gait cycles. Due to these concerns, and the difficulty of accurately modeling robot and human interactions and the requirement for online human intention prediction, RL has emerged as a viable option.

Recent research on power augmentation in exoskeletons has utilized a variety of RL-based techniques. A Hierarchical Interactive Learning (HIL) controller was used to manage changing interaction dynamics [1]. Motion trajectories were modeled using Dynamical Movement Primitives (DMP) and learnt with the Locally Weighted Regression (LWR) technique in high-level motion learning. The controller in the low-level hierarchy was learned using an RL technique named Q-learning. Experiments were carried out on a platform with a single degree of freedom (DOF) and a lower-limb exoskeleton. When compared to a conventional model-based controller, the results indicated enhanced control performance.

An Interactive Learning Actor-Critic (ILAC) controller was suggested in [2] to address dimensionality problems in continuous observation spaces. The sensitivity factor and the compensatory coefficients were learned using the actor-critic method. Both a single DOF device and a lower-limb exoskeleton were used in the experiments. The findings revealed that continuous high-dimensional observation spaces performed better than discretized low-dimensional observation spaces while maintaining a small error.

A hierarchical learning system was developed in [3], consisting of a high-level motion learning and a low-level impedance controller. Rhythmical Movement Primitives (RMP) were used to represent human gait. The policy parameters were then optimized using a reinforcement learning technique called Policy Improvement with Path Integrals and Covariance Matrix Adaptation Evolution Strategy (PI^2-CMA-ES). Experiments on a single DOF as well as a lower-limb exoskeleton were carried out. Compared to current model-based controllers, the results showed that the approach could decrease Human-Machine Interaction (HMI).

Coupled Cooperative Primitives (CCP) were used in [4] to learn real-time gait trajectories. The CCP utilized an HMI model with modulation terms. The PI^2 technique optimized the modulation settings in the situation of changing HMI. Experiments with a single DOF device and an exoskeleton were conducted. The outcomes of the experiments demonstrated that the suggested approach outperformed conventional representations.

A gait planner based on DMPs and RL was employed in [5] to extract the user movement intention in uphill slope walking. A hybrid HMI model was created using the Center of Mass (CoM) of a user with paraplegia. The model's

978-1-6654-2094-5/21/$31.00 ©2021 IEEE

parameters were then learned using the PI^2 approach. The suggested strategy was tested, and the results showed that it improves the user's stability and comfort while walking uphill slopes.

For a single DOF rehabilitation ankle-foot orthosis, an Assist-As-Needed (AAN) controller based on RL was suggested in [6]. The actor-critic algorithm was used to assess the user's performance and arrive at the best control policy. Furthermore, Action Dependent Heuristic Dynamic Programming (ADHDP) was employed to remove the requirement for a system model. With four healthy volunteers, experiments were performed on a single DOF ankle-foot orthosis. According to the results, because of its higher effectiveness compared to traditional controllers, the suggested control method may be utilized for rehabilitation activities.

RL-based control methods often suffer from the curse of dimensionality in high-dimensional continuous learning spaces, resulting in delayed convergence and poor performance. Furthermore, safety is a major concern in situations where the RL algorithm must interact with a person. As a result, measures should be taken throughout the learning phase to guarantee that the RL algorithm explores safely and efficiently. This implies that the RL algorithm should not visit unsafe states or take risky actions during exploration.

Some new studies on RL-based robot skill learning have used dimensionality reduction methods to create low-dimensional latent parameter spaces [7]. To begin, the DMP parameter space was reduced using principal component analysis. Second, the latent space was defined using a deep AE neural network. A robotic arm was used to test the methods in a ball throwing job. Experiments have shown that the generated latent space allows RL techniques to converge more quickly. However, to the best of our knowledge, the use of AEs in lower-limb exoskeletons for safe RL and rapid convergence has not been investigated.

This paper proposes a control strategy based on safe RL to detect human motion intention and reduce HMI forces. To this end, DMPs are used to describe the policy for RL and are utilized to parameterize gait trajectories. A deep neural network structure called AE is also used to reduce dimensionality and generate a latent space of valid trajectories. Compared to traditional approaches, this allows for safer exploration, faster convergence, and higher control performance.

The layout of this paper is presented below. The materials and methods are discussed in Section II. The simulation results are shown in Section III. In Section IV, experimental results are provided. Lastly, a conclusion is reached in Section V.

II. Materials and Methods

A. Autoencoder training and dimensionality reduction

First, we will go through the suggested approach for resolving the aforementioned issue. We begin by describing the learning space and then discuss how to use AEs to reduce the learning space's dimensionality. The use of statistical generalization techniques to database development is also discussed. We start with a small collection of real-world tasks and expand from there. The AE is then trained using the augmented dataset. Because the AE's latent space is smaller, RL can converge rapidly and safely. Because the data for AE training is based on a limited collection of real-world activities, it is also more realistic than an AE trained solely in the simulation environment. Since RL is conducted in a smaller space specified by the latent space of the AE, the result is that we learn quicker and safer.

At first, a parametric policy $\Pi(\boldsymbol{\theta})$ is considered, where $\boldsymbol{\theta} \in \Theta$ denotes the policy parameters and $\boldsymbol{\theta} \subset \mathbb{R}^d$ specifies the space of all potential policy parameters. In a properly created lower-dimensional parameter space, learning will be simpler, faster, and safer. To accomplish this, AE-based dimensionality reduction is suggested, which blends all feasible task policies into a lower-dimensional latent space.

As demonstrated in [7], Deep AE neural networks may reduce dimensionality while preserving the essential information in the lower dimensions. The AE is taught to match inputs and outputs exactly. The inputs move through the network and arrive at the latent space layer with the fewest neurons. The values of these neurons are denoted by $\boldsymbol{\theta}_{AE}$. The encoder portion runs from the input to the latent layer, while the decoder moves the data through increasingly larger layers such that the AE's outputs $\tilde{\boldsymbol{\theta}}$ resemble the inputs $\boldsymbol{\theta}$ as accurately as feasible. The corresponding functions of encoder and decoder are approximately inverse $F_{dec} \approx F_{enc}^{-1}$. The latent space now defines data dimensionality, which is typically smaller than the original space's dimensionality.

In order to train the AE neural network, the averaged Euclidian distance between the AE's inputs and outputs is minimized across all sample trajectories in the training set via the stochastic gradient descent approach. After the learning is complete, the latent parameters can be computed as:

$$\boldsymbol{\theta}^{AE} = \mathbf{F}_{enc}(\boldsymbol{\theta}) \tag{1}$$

Additionally, the original policy parameters are calculated as follows:

$$\tilde{\boldsymbol{\theta}} = \mathbf{F}_{dec}(\boldsymbol{\theta}^{AE}) \tag{2}$$

B. Statistical dataset augmentation

AE, like other neural networks, requires a large number of data to train. Obtaining such a large number of data points is an arduous task that puts the equipment under excessive pressure and may potentially harm the robot and the user. Simulated data collection is one approach to solve these problems. However, a mismatch exists between simulation and real-world dynamics. One of the contributions of this study is a method for synthesizing training samples without the requirement for real data gathering. To compute the AE network defining the AE-based latent space, a dataset of real-world trajectories is gathered $\{\boldsymbol{\theta}_i\}_{i=1}^{N}$. This article offers a novel technique for generating data for learning AE utilizing (GPR) [8]. The aim is to gather a limited number of real trajectories, then use GPR to create a larger collection of artificial data based on the actual data. We chose GPR over other statistical learning techniques because it has been shown [8] that GPR outperforms other methods in various real situations.

Assume the robot completed a limited amount of tasks. The related motions and task descriptions are referred to as $D=\{\mathbf{q}_i,\boldsymbol{\theta}_i^{DMP}\}$ as $\boldsymbol{\theta}_i^{DMP}$ and $\mathbf{q}_i$, respectively. We create a dataset:

$$D=\{\mathbf{q}_i,\boldsymbol{\theta}_i^{DMP}\} \tag{3}$$

By choosing a new query point and calculating the corresponding motion policy using GPR, new data points may be generated:

$$gp:\mathbf{q}_d \rightarrow \boldsymbol{\theta}_d^{DMP} \tag{4}$$

Our hypothesis, which has been validated in our tests, is that the data produced using this method will perform as well as or better than simulation data. As a result of a better starting approximation, the RL will become faster.

C. Reinforcement Learning

One of the most important issues of learning in large continuous state-action spaces is the curse of dimensionality. To address this, the PI^2-CMA algorithm employs a parametric policy and probability-weighted averaging [9]. First, a set of k rollouts are chosen from a Gaussian distribution. $\boldsymbol{\theta}$ denotes policy parameter vector which produces the trajectory denoted by $(\tau_{i=1,\ldots,n})$. The algorithm searches the policy parameter space, and the cost function is computed during trajectory tracking. The cost of the i-th time step is represented by J_i. Incorporating these costs, the cost of executing a rollout will be calculated. Then appropriate probability values are assigned according to the cost of each rollout so that less costly rollouts are given more importance. A newly updated parameter vector $(\boldsymbol{\theta}_i^{new})$ is calculated at each time step. The final updated parameter is calculated using temporal averaging. The pseudocode of the PI^2-CMA algorithm is presented in Algorithm 1.

D. Control Strategy Overview

To perform the GPR-based dataset augmentation detailed in Section II, Comprehensive query points are needed to form a well-defined trajectory dataset. Two gait features are used as query points, namely amplitude and frequency. To extract these features, a pool of adaptive frequency oscillators (AFO) was used [10]. Furthermore, the deep AE network is trained in an offline manner.

An overview of the overall control strategy deployed in real-time is provided in Fig. 1. Firstly, the amplitude and frequency are extracted as gait features using the AFOs. Secondly, these query points are fed to the GPR to extract initial estimates for the DMP policy parameters. Thirdly, dimensionality reduction is performed via the encoder part of the AE to calculate the initial latent parameters to be optimized using RL. Then, the PI^2-CMA algorithm performs RL in the latent space in a faster and safer manner. After the optimal latent parameters are found, the decoder part of the AE transforms them into the corresponding optimal DMP policy parameters, which is converted into the optimal trajectory to be tracked by the low-level controller. Finally, a Time Delay Estimation Continuous Fractional Order Nonsingular Terminal Sliding Mode (TDE-CFONTSM) Controller is used as the low-level controller to track the desired trajectory provided by RL. The details of the TDE-CFONTSM controller and the stability analysis are provided in [11].

Algorithm 1 PI^2-CMA algorithm

trials→ for k=1,...,K:

sampling→ $\quad\boldsymbol{\theta}_k \sim N(\boldsymbol{\theta},\Sigma)$

policy→ $\quad\boldsymbol{\tau}_{k,i=1,\ldots,N}=\text{policy}(\boldsymbol{\theta}_{k,i=1,\ldots,N})$

time steps→ for i=1,...,N:

trials→ $\quad$ for k=1,...,K:

evaluate→ $\quad S_{k,i}\equiv S(\boldsymbol{\tau}_{k,i})=\sum_{j=i}^{N} J(\boldsymbol{\tau}_{j,k})$

probability→ $\quad P_{k,i}=\dfrac{e^{-\frac{1}{\lambda}S_{k,i}}}{\sum_{k=1}^{K}\left[e^{-\frac{1}{\lambda}S_{k,i}}\right]}$

update→ $\quad\boldsymbol{\theta}_i^{new}=\sum_{k=1}^{K} P_{k,i}\boldsymbol{\theta}_k$

Covar. adap.→ $\quad\Sigma_i^{new}=\sum_{k=1}^{K} P_k(\boldsymbol{\theta}_{k,i}-\boldsymbol{\theta})(\boldsymbol{\theta}_{k,i}-\boldsymbol{\theta})^T$

Temp. avg.→ $\quad\boldsymbol{\theta}_{new}=\dfrac{\sum_{i=0}^{N}(N-i)\boldsymbol{\theta}_i^{new}}{\sum_{l=0}^{N}(N-l)}$

Covar. avg.→ $\quad\Sigma_{new}=\dfrac{\sum_{i=0}^{N}(N-i)\Sigma_i^{new}}{\sum_{l=0}^{N}(N-l)}$

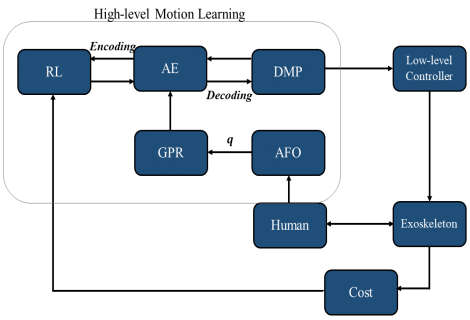

Fig. 1. Structure of the proposed controller

III. Simulation Results

In the following, the parameters of each module used in the proposed control strategy is provided. The structure of the deep AE network is presented in Fig. 2.

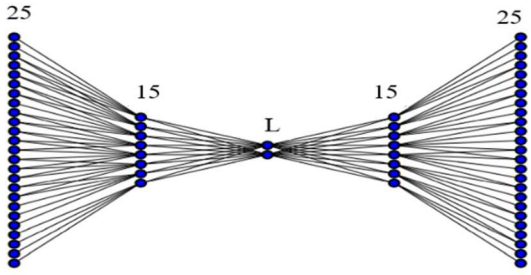

Fig. 2. Structure of the deep AE network

In order to choose the minimum latent space dimension denoted by L, simulations were carried out using different values of L. The results are presented in Table I.

TABLE I. AE PERFORMANCE BASED ON LATENT SPACE SIZE

L	2	4	6	8	10
MSE_θ	0.0454	0.0195	0.0162	0.0114	0.0057

As seen in the table above, as L decreases, more information is lost when data is passed through the AE, which is undesirable. To answer this trade-off, $L = 4$ was chosen for the remainder of the experiments.

As for the RL algorithm, the cost function was considered as:

$$r(t) = \beta_1(\theta_h - \theta_e)^2 + \beta_2(\dot{\theta}_h - \dot{\theta}_e)^2 + \beta_3 F_{\text{int}}^2 \qquad (5)$$

The convergence criteria were defined as $R<\eta$, wherein R is the cost to execute a trajectory.

Finally, the parameters of the low-level TDE-CFONTSM controller are given in Table II.

TABLE II. PARAMETERS OF THE PROPOSED CONTROL STRATEGY

TDE-CFONTSM		**DMP**	
Parameter	Value	Parameter	Value
k	1	N	25
k_1	10		
k_2	0.5	α_z	4
α	0.8		
α_k	1	β_z	1
b	0.8		
λ_1	0.9	g	0
ρ	4		
δ_0	0.2	λ_2	0.99

The simulations were conducted in the MATLAB/Simulink environment. In the following, the simulation results are presented:

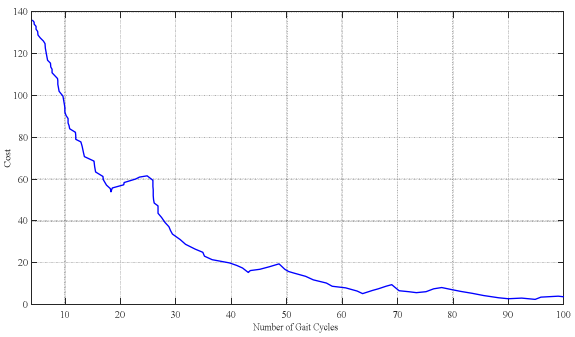

Fig. 3. Cost curve of the proposed control strategy during simulation

As presented in Fig. 3, the RL algorithm converges after approximately 90 gait cycles, which means the latent parameters are updated 18 times, as shown in Fig. 4.

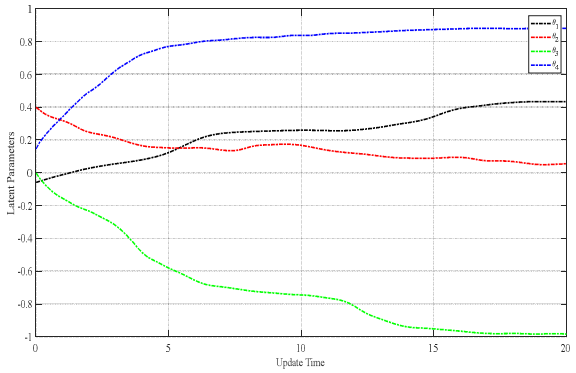

Fig. 4. Learning of latent policy parameters during simulation

Furthermore, trajectory tracking performance is presented in Fig. 5.

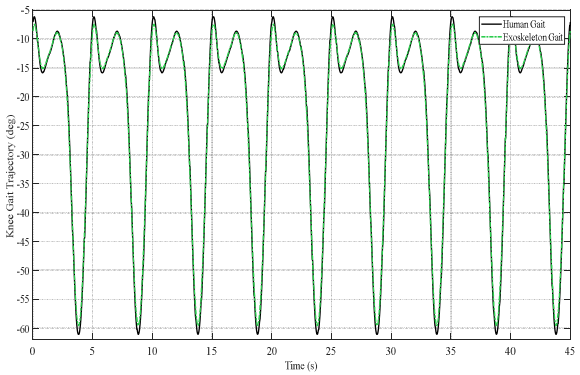

Fig. 5. Tracking performance of the proposed strategy during simulation

It is evident from the figure above that the RL algorithm successfully estimates and tracks the user's motion. The RMS of tracking error is 0.81 (deg).

IV. EXPERIMENTAL RESULTS

A. Apparatus

The experiments were carried out using the Sharif Powered Exoskeleton (SPEX). Further details and design specifications are provided in [12]. A general overview of the SPEX robot's structure is presented in Fig. 6.

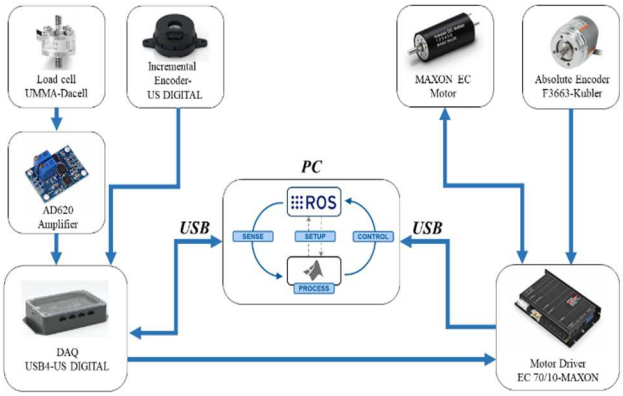

Fig. 6. The overall structure of the SPEX robot

B. *Experimental protocols*

To ensure the validity of the experimental results and prevent user bias towards the robot, a unique protocol was implemented, as detailed below:

- The user trained for a week to perform the single DOF knee gait according to external cues provided by a metronome.
- The frequency of the movement was kept constant to prevent the user from adapting to the robot instead of the other way around.
- The amplitude of the gait cycle varied between different cycles, as expected in natural walking.
- Upon divergence according to the criterion, the robot extracted the gait features and performed RL in zero-force mode to avoid excessive interaction force until convergence.

C. *Results*

Two experiments were conducted on a healthy adult with two different frequencies to verify the ability of the RL-based control strategy to adapt to different walking patterns. The first experiment was conducted at a slow pace ($f = 0.1\ (Hz)$), and the second experiment was done at a moderate pace ($f = 0.2\ (Hz)$) with the aforementioned protocol. The results of the first experiment are presented as follows.

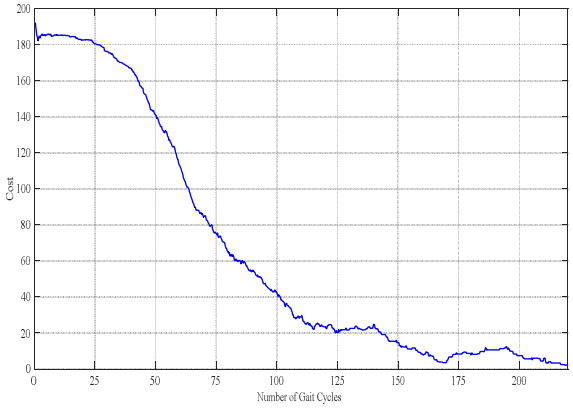

Fig. 7. Cost curve of the proposed control strategy during learning for the first experiment

According to Fig. 7, the algorithm converges after roughly 190 cycles. During the learning phase, the latent parameters are updated 38 times, as shown in Fig. 8.

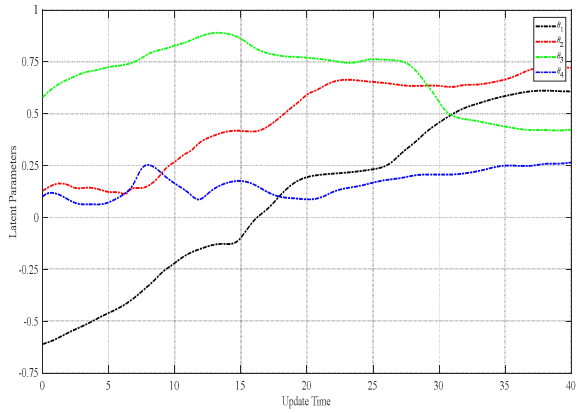

Fig. 8. Learning of latent policy parameters for the first experiment

Furthermore, Fig. 9 demonstrates the tracking performance for the first experiment.

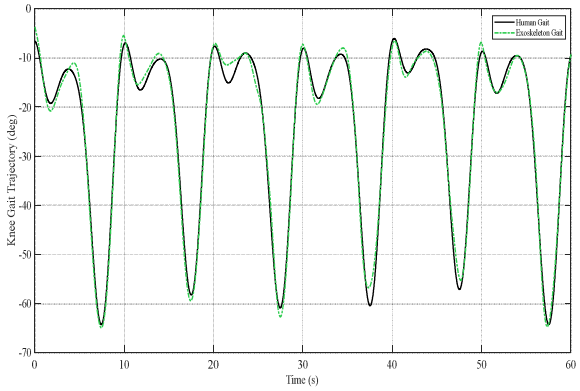

Fig. 9. Tracking performance of the proposed strategy in the first experiment

It can be concluded from Fig. 9 that the control strategy has been able to synchronize itself with the user successfully. The RMS of tracking error is 1.43 (deg). The interaction force measured by force sensors is also given in Fig. 10.

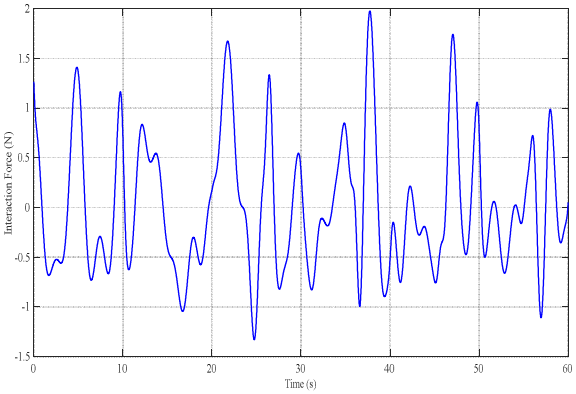

Fig. 10. The interaction force between robot and human in the first experiment

As seen in Fig. 10, the interaction force has been managed successfully and kept within the range of $[-2,2]\ (N)$.The RMS of interaction force is 1.05 (N).

Subsequently, the results of the second experiment are provided.

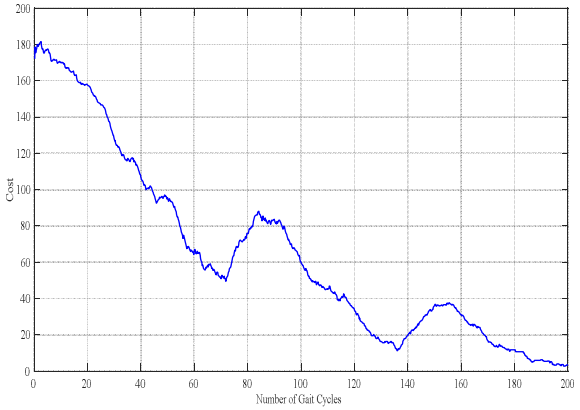

Fig. 11.Cost curve of the proposed control strategy during learning for the second experiment

It can be inferred from Fig. 11 that convergence has been achieved after approximately 200 cycles. This means that the latent policy parameters are updated around 40 times, as shown in Fig. 12.

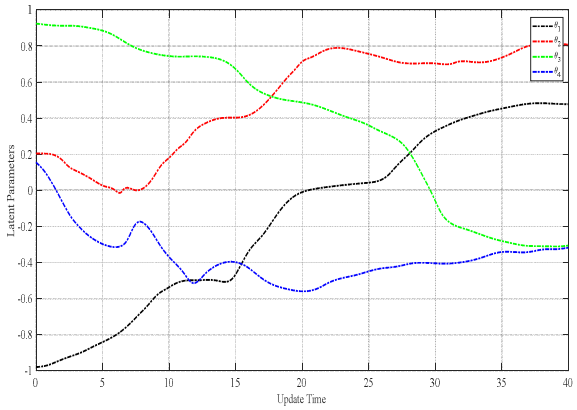

Fig. 12. Learning of the latent policy parameters for the second experiment

Tracking performance for the second experiment is provided in Fig. 13.

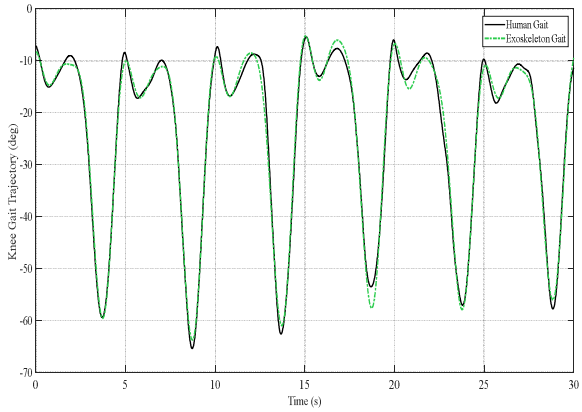

Fig. 13. Tracking performance of the proposed strategy in the second experiment

According to Fig. 13, good trajectory tracking performance has been achieved with the RMS of tracking error equal to 1.95 (deg). The interaction force has been kept between -2 and +2 (*N*) for the most part, as presented in Fig. 14.

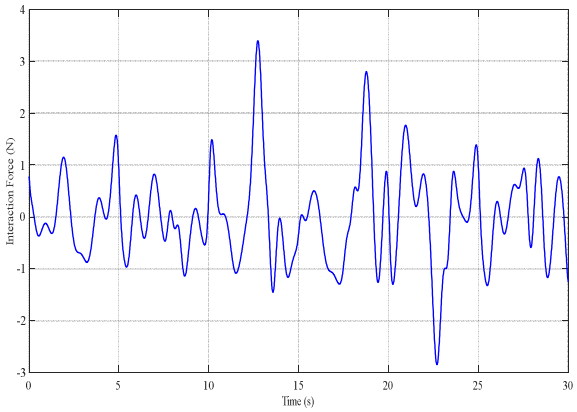

Fig. 14. The interaction force between robot and human in the second experiment

According to Fig. 14, the RMS of interaction force is 1.49 (N).

Finally, the results have been compared to that of the SAC and HIL, which are model-based and learning-based strategies [13], [1].

TABLE III. COMPARISON OF TRACKING PERFORMANCE WITH CONVENTIONAL METHODS

Control Strategies	Experiment 1 (deg)	Experiment 2 (deg)
HIL	1.55	2.07
SAC	2.76	4.93
Current study	1.44	1.95

REFERENCES

[1] R. Huang, H. Cheng, H. Guo, X. Lin, J. Zhang, "Hierarchical learning control with physical human-exoskeleton interaction," Information Sciences, vol. 432, pp. 584-595, 2018.

[2] G. Song, R. Huang, H. Cheng and Q. Chen, "Learning Coupled Parameters with Continuous Domains for Human-powered Lower Exoskeleton," 2018 3rd International Conference on Advanced Robotics and Mechatronics (ICARM), pp. 189-194, 2018.

[3] L. Wang, Z. Du, W. Dong, et al., "Hierarchical Human Machine Interaction Learning for a Lower Extremity Augmentation Device," Int. J. of Soc. Robotics, vol. 11, pp. 123–139, 2019.

[4] R. Huang, H. Cheng, J. Qiu, J. Zhang, "Learning Physical Human–Robot Interaction With Coupled Cooperative Primitives for a Lower Exoskeleton," in IEEE Transactions on Automation Science and Engineering, vol. 16, no. 4, pp. 1566-1574, 2019.

[5] R. Huang, Q. Wu, J. Qiu, H. Cheng, Q. Chen, Z. Peng, "Adaptive Gait Planning with Dynamic Movement Primitives for Walking Assistance Lower Exoskeleton in Uphill Slopes," Sensors and Materials, vol. *32, no.* 4, pp. 1279-1291, 2020.

[6] Y. Zhang, S. Li, K. J. Nolan and D. Zanotto, "Adaptive Assist-as-needed Control Based on Actor-Critic Reinforcement Learning," 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4066-4071, 2019.

[7] R. Pahič, Z. Lončarević, A. Gams, A. Ude, "Robot skill learning in latent space of a deep autoencoder neural network," Robotics and Autonomous Systems, vol. 135, 2021.

[8] C. Rasmussen, C. Williams, "Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)," Cambridge, MA, USA: The MIT Press, 2005, pp. 69-106.

[9] F. Stulp, O. Sigaud, "Path Integral Policy Improvement with Covariance Matrix Adaptation," Proceedings of the 10th European Workshop on Reinforcement Learning (EWRL 2012), 2012.

[10] H. Talatian, M. Karami, H. Moradi and G. Vossoughi, "Design and Implementation of an Intelligent Control System for a Lower-Limb Exoskeleton to Reduce Human Energy Consumption," 2021 10th International Conference on Modern Circuits and Systems Technologies (MOCAST), pp. 1-4, 2021.

[11] M. Karami, M. Abbasi and G. Vossoughi, "Design of a Continuous Fractional-Order Nonsingular Terminal Sliding Mode Control with Time Delay Estimation for FES Control of Human Knee Joint," 2021 7th International Conference on Control, Instrumentation and Automation (ICCIA), pp. 1-5, 2021.

[12] A. Taherifar, et al., "Design and control of an assistive exoskeleton with passive toe joint." International Journal of Mechatronics and Automation, vol. 6. pp. 83-93, 2018.

[13] H. Kazerooni, J. Racine, L. Huang, R. Steger, "On the Control of the Berkeley Lower Extremity Exoskeleton (BLEEX)," Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pp. 4353-4360, 2005.