Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Probabilistic movement primitive based motion learning for a lower limb exoskeleton with black-box optimization*

Jiaqi WANG, Yongzhuo GAO, Dongmei WU, Wei DONG‡

State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, Harbin 150001, China

E-mail: wangjq@hit.edu.cn; gaoyongzhuo@hit.edu.cn; wdm@hit.edu.cn; dongwei@hit.edu.cn

Received Feb. 22, 2022; Revision accepted Aug. 8, 2022; Crosschecked Oct. 20, 2022

Abstract: As a wearable robot, an exoskeleton provides a direct transfer of mechanical power to assist or augment the wearer's movement with an anthropomorphic configuration. When an exoskeleton is used to facilitate the wearer's movement, a motion generation process often plays an important role in high-level control. One of the main challenges in this area is to generate in real time a reference trajectory that is parallel with human intention and can adapt to different situations. In this paper, we first describe a novel motion modeling method based on probabilistic movement primitive (ProMP) for a lower limb exoskeleton, which is a new and powerful representative tool for generating motion trajectories. To adapt the trajectory to different situations when the exoskeleton is used by different wearers, we propose a novel motion learning scheme based on black-box optimization (BBO) PI^{BB} combined with ProMP. The motion model is first learned by ProMP offline, which can generate reference trajectories for use by exoskeleton controllers online. PI^{BB} is adopted to learn and update the model for online trajectory generation, which provides the capability of adaptation of the system and eliminates the effects of uncertainties. Simulations and experiments involving six subjects using the lower limb exoskeleton HEXO demonstrate the effectiveness of the proposed methods.

Key words: Lower limb exoskeleton; Human-robot interaction; Motion learning; Trajectory generation; Movement primitive; Black-box optimization

https://doi.org/10.1631/FITEE.2200065 **CLC number:** TP242.6

1 Introduction

In recent years, there has been growing interest in the application of robotic exoskeletons as a solution to assist people undertaking activities. Many lower extremity exoskeletons have been developed and are widely used in power augmentation (Guizzo and Goldstein, 2005; Kazerooni and Steger, 2006; Zoss et al., 2006; Walsh et al., 2007), walking assistance (Sankai, 2010; Hassan et al., 2014), and rehabilitation training (Colombo et al., 2000; Veneman et al., 2007; Strausser and Kazerooni, 2011; Esquenazi et al., 2012; Sanz-Merodio et al., 2014). One of the toughest issues in this area is that, as a typical human-robot coupling system, the exoskeleton should work cooperatively with the human wearer (Deng et al., 2020). Though the development of appropriate control strategies is fast, motion learning is one of the main research subjects in the field of exoskeleton robots (Lee et al., 2015; Yan et al., 2015; Xu and Sun, 2018). A human-like reference trajectory can help the exoskeleton system achieve favorable human-robot interaction and is directly related to the comfort of the wearer. In addition, trajectory in parallel with human intention is considered to help smooth the movement and optimize the mechanical efficiency to save energy.

Motion trajectory generation has been extensively investigated by researchers in the field of human-robot

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. U21A20120)

ORCID: Jiaqi WANG, https://orcid.org/0000-0003-4084-135X; Wei DONG, https://orcid.org/0000-0002-1211-6444

interaction. A model-based strategy is a classic method to generate trajectories for lower limbs (Kagawa et al., 2015; Kazemi and Ozgoli, 2019). Based on the model and the stability criteria, like the link model (Fu and Chen, 2008), inverted pendulum model (IPM) (Komura et al., 2005), and zero-moment point (ZMP) model (Vukobratović and Borovac, 2004; Al-Shuka et al., 2016; He et al., 2017), the trajectories are generated using mathematical expressions. This kind of method relies on the accuracy of the human-exoskeleton and environment model, so its effectiveness is limited by this objective condition (Kazemi and Ozgoli, 2019). Therefore, the trajectory has poor adaptability to the actual environment and poor robustness to disturbance.

The role of the exoskeleton is to provide walking assistance in coordination with a person. Naturally, exoskeletons generate trajectories by emulating human movement. It is essential to detect and realize the wearer's movement instead of using a predefined motion. Some intelligent technologies have been developed for this, and learning from demonstration has recently gained considerable interest in studies of robot systems (Yang et al., 2019; Deng et al., 2020). Movement primitive (MP) is a well-established approach for representing and generating movement from demonstration (Schaal et al., 2003; Krüger et al., 2007; Kulić et al., 2012). Ijspeert et al. (2002, 2013) proposed a tool named dynamic movement primitive (DMP) for representing rhythmic and discrete trajectories. In Huang et al. (2018), DMP was combined with locally weighted regression (LWR) to model exoskeleton trajectories. Representing motion by means of MP is considered motion generation. Continuous learning is also required to achieve the flexibility needed in a human-robot coupling system. An exoskeleton with self-adaptive motion learning is adaptable to different wearers and environments, and can reduce the effect of uncertainties and disturbances. Yuan et al. (2020) proposed a trajectory-learning scheme for motion generation based on path integrals (PI^2) combined with DMP. Huang et al. (2019) proposed coupled cooperative primitives (based on DMP) to learn the motion, using policy improvement with PI^2 to update the parameters. Their results, and those of other studies, demonstrated the stable performance of the system after the motion learning converged. However, too many iterations are needed in the learning process. It may not be a serious issue if the exoskeleton maintains a steady walking pace, but this learning ability deals poorly with complex and variable walking situations. Besides, the need for almost 30 iterations every time the subject is changed is a challenge.

Almost all the motion generations of the lower limb exoskeleton can be learned by PI^2, because PI^2 is an efficient and easy-to-implement algorithm of reinforcement learning (RL) (Theodorou et al., 2010; Schmidhuber, 2015). However, to further improve the performance, a more efficient algorithm is needed. We have shown that the algorithm PI^{BB}, devised by Stulp and Sigaud (2012), outperforms PI^2 in terms of convergence speed and final cost. As a modification of PI^2, PI^{BB} simplifies the exploration and parameter update methods of PI^2. In essence, PI^{BB} is a kind of black-box optimization (BBO) algorithm.

The convergence speed depends not only on the learning algorithm, but also on the representative ability of MP. DMP is a commonly used trajectory-based representation approach (Yang et al., 2019). However, DMP is more suitable for learning a point-to-point trajectory because of the convergence nature of its attractor. Generalization to new and unseen situations in DMPs is limited, so further work is needed for representing optimal behaviors (Stulp and Sigaud, 2012). Moreover, traditional MP may have the problems of a relatively low speed and low accuracy. Therefore, in this paper, some new MP concepts are presented and implemented. A novel probabilistic movement primitive (ProMP) was proposed by Paraschos et al. (2013, 2018). In ProMP, probability distribution is used to encode the movement, as it is often a requirement for representing optimal behaviors. In contrast, deterministic approaches such as DMP can represent only the suboptimal ones. Moreover, capturing the variance of the movement leads to better generalization capabilities (Todorov and Jordan, 2002; Schaal et al., 2005). Most importantly, unlike past approaches (d'Avella and Bizzi, 2005) to learn movements from a single demonstration, ProMP can be learned from multiple demonstrations by incorporating the variance. This increases flexibility and enhances the advantageous properties of the representation. For exoskeletons, this kind of representation learned from multiple motion habits can generate a more general human motion trajectory.

In this paper, we propose a novel motion learning scheme for a lower limb exoskeleton. For motion generation, a powerful motion representative tool, ProMP, is used to model exoskeleton motion trajectories from multiple demonstrations. To our knowledge, ProMP has not previously been used for motion planning of an exoskeleton. Then, for motion adaptation, the optimization algorithm PI^{BB} is adopted to learn and update the ProMP motion model online, so the exoskeleton can adapt to different wearers and variable environments. Simulations and experiments demonstrate the effectiveness of the proposed methods. The motion learning can quickly adapt to a new wearer and generate a trajectory in parallel with human intention. The convergence speed is higher than that of the existing methods. The human-exoskeleton system can obtain better flexibility and faster movement coordination.

2 Method

Fig. 1 shows the working framework of the exoskeleton system with the proposed motion learning strategy. The major subsystems in this generalized framework include a hierarchical control structure (Tucker et al., 2015), the wearer, and the exoskeleton. The proposed motion learning strategy occupies a high-level layer of the structure as shown in the solid box, consisting of initial motion generation shown by the dotted line, and motion adaptation shown by the dashed box. The motion model is first generated offline from trajectory demonstration, and then updated online by an optimization algorithm. The trajectory learned offline is regarded as the reference trajectory for online working. The trajectory is changed and updated when the subject begins to move. The data from the actual joint trajectories are used in the PI^{BB} algorithm to calculate the corresponding cost value, and then the parameters are updated. Next, the ProMP algorithm with the updated parameters generates new desired trajectories for the lower limb exoskeleton. In the following subsections, we explain how the motion is represented and learned by ProMP, and provide details about the trajectory adaptation.

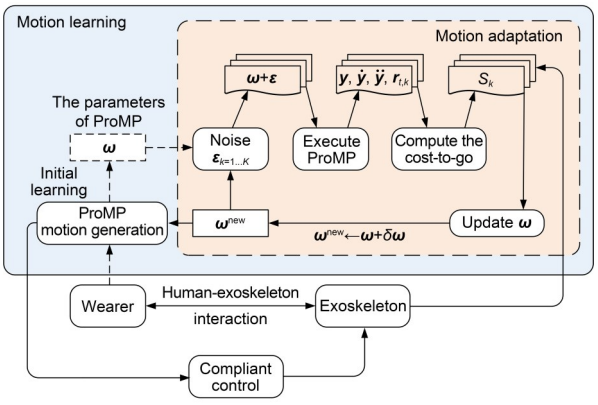

Fig. 1 The working framework of the exoskeleton system with the proposed motion learning strategy

2.1 Motion representation

In terms of ProMP, a probabilistic model based on the basis function is introduced to represent the trajectory. The trajectory distribution of the lower limb exoskeleton in this research focuses on the joint space. q_t and $\dot{q}_t$ are used to represent the joint angular position and joint angular velocity, respectively, of each degree of freedom (DOF) at time t. $\boldsymbol{\omega}$ is used to produce a single trajectory as an underlying weight vector. A linear basis function is used to model the state of the joint:

$$\boldsymbol{y}_t = \begin{bmatrix} q_t \\ \dot{q}_t \end{bmatrix} = \boldsymbol{\Phi}_t^{\mathrm{T}}(\boldsymbol{\omega} + \boldsymbol{\varepsilon}_t), \tag{1}$$

where $\boldsymbol{\Phi}_t = [\boldsymbol{\phi}_t, \boldsymbol{\phi}_t]$ is the $N\times 2$ dimensional time-dependent basis function matrix, and N defines the number of basis functions. $\boldsymbol{\varepsilon}_t \sim \mathcal{N}(0, \boldsymbol{\Sigma}_t)$ is Gaussian noise with 0 mean.

Based on Spiegelhalter et al. (2002) and Tucker et al. (2015), with $\boldsymbol{\omega}$ maintaining a Gaussian distribution $\boldsymbol{\omega} \sim \boldsymbol{p}(\boldsymbol{\omega}; \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\omega} | \boldsymbol{\mu}_\omega, \boldsymbol{\Sigma}_\omega)$ with parameters $\boldsymbol{\theta}$, the trajectory distribution is introduced as

$$\boldsymbol{p}(\boldsymbol{y}_t; \boldsymbol{\theta}) = \int \boldsymbol{p}(\boldsymbol{y}_t | \boldsymbol{\omega})\, \boldsymbol{p}(\boldsymbol{\omega}; \boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\omega}. \tag{2}$$

The distribution $\boldsymbol{p}(\boldsymbol{y}_t; \boldsymbol{\theta})$ defines the hierarchical Bayesian model whose parameters are given by the parameters $\boldsymbol{\theta}$ and the observation noise variance $\boldsymbol{\Sigma}_y$.

Temporal modulation is needed for adapting to changes in walking speed. A phase variable z is introduced to separate the movement from the time signal. The phase can be any function that monotonically increases with time t, and the speed of the movement

can be modulated by modifying the rate of the phase variable α. In this study, z_t is adopted as

$$z_t = \alpha t. \tag{3}$$

At the beginning of the movement, the phase z_0 is defined as 0, and in the end, the phase is $z_E = 1$. The basis function $\boldsymbol{\phi}_t$ now directly depends on the phase instead of the time:

$$\boldsymbol{\phi}_t = \boldsymbol{\phi}(z_t). \tag{4}$$

2.2 Motion initial learning

The probabilistic model represents the trajectory distribution based on a basis function. For human walking motion, the Von-Mises basis functions $\boldsymbol{b}_i$ (Jenison and Fissell, 1995) for rhythmic movement are used to model periodicity in the phase variable z:

$$\boldsymbol{b}_i(z_t) = \exp\left(\frac{\cos\left(2\pi(z_t - c_i)\right)}{h}\right), \tag{5}$$

where h denotes the width of the basis, and c_i is the center of the i^{th} basis function. Then, it is normalized by

$$\boldsymbol{\phi}_i(z_t) = \frac{\boldsymbol{b}_i(z_t)}{\sum_{j=1}^{N}\boldsymbol{b}_j(z_t)}. \tag{6}$$

The distribution $\boldsymbol{p}(\boldsymbol{y}_t; \boldsymbol{\theta})$ for time step t is shown in Eq. (7), by which the mean and the variance for any time point t can be evaluated:

$$\begin{aligned}\boldsymbol{p}(\boldsymbol{y}_t; \boldsymbol{\theta}) &= \int \mathcal{N}\left(\boldsymbol{y}_t|\boldsymbol{\Phi}_t^{\mathrm{T}}\boldsymbol{\omega}, \boldsymbol{\Sigma}_y\right)\mathcal{N}\left(\boldsymbol{\omega}|\boldsymbol{\mu}_\omega, \boldsymbol{\Sigma}_\omega\right)\mathrm{d}\boldsymbol{\omega}\\ &= \mathcal{N}\left(\boldsymbol{y}_t|\boldsymbol{\Phi}_t^{\mathrm{T}}\boldsymbol{\mu}_\omega, \boldsymbol{\Phi}_t\boldsymbol{\Sigma}_\omega\boldsymbol{\Phi}_t^{\mathrm{T}} + \boldsymbol{\Sigma}_y\right).\end{aligned} \tag{7}$$

To generate motion, $\boldsymbol{p}(\boldsymbol{\omega};\boldsymbol{\theta})$ needs to be learned from multiple demonstrations. Assuming that there are M demonstration trajectories, the weight for each trajectory is estimated using linear ridge regression:

$$\boldsymbol{\omega}_m = \left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y}_m, \tag{8}$$

where $\boldsymbol{Y}_m$ represents the position of all steps for the m^{th} demonstration trajectory, and λ is a regression parameter. Then the parameters $\boldsymbol{\theta} = \{\boldsymbol{\mu}_\omega, \boldsymbol{\Sigma}_\omega\}$ are obtained using the maximum likelihood estimation algorithm. The mean $\boldsymbol{\mu}_\omega$ and covariance $\boldsymbol{\Sigma}_\omega$ are computed from samples $\boldsymbol{\omega}_m$:

$$\begin{cases}\boldsymbol{\mu}_\omega = \dfrac{1}{M}\sum_{m=1}^{M}\boldsymbol{\omega}_m,\\ \boldsymbol{\Sigma}_\omega = \dfrac{1}{M-1}\sum_{m=1}^{M}(\boldsymbol{\omega}_m - \boldsymbol{\mu}_\omega)(\boldsymbol{\omega}_m - \boldsymbol{\mu}_\omega)^{\mathrm{T}}.\end{cases} \tag{9}$$

2.3 Motion adaptation

The optimization algorithm adopted in this study is $\mathrm{PI}^{\mathrm{BB}}$. This kind of policy improvement algorithm is updated based on each improved execution or "roll-out." Based on a total of K alternative trajectories with slight differences, policy improvement methods then update the parameter vector $\boldsymbol{\omega} \rightarrow \boldsymbol{\omega}^{\text{new}}$ such that the policy is expected to incur lower costs.

The policy perturbation during a roll-out is generated from the model of the trajectory with noise

$$\boldsymbol{y}_t = \boldsymbol{\Phi}_t^{\mathrm{T}}(\boldsymbol{\omega} + \boldsymbol{\varepsilon}). \tag{10}$$

Then, based on Paraschos et al. (2018), the cost function formula in the roll-out policy of each of the k^{th} roll-out trajectories is

$$\boldsymbol{M}_{t,k} = \frac{\boldsymbol{J}^{-1}\boldsymbol{\Phi}_{t,k}\boldsymbol{\Phi}_{t,k}^{\mathrm{T}}}{\boldsymbol{\Phi}_{t,k}^{\mathrm{T}}\boldsymbol{J}^{-1}\boldsymbol{\Phi}_{t,k}}, \tag{11}$$

$$\boldsymbol{S}_k = \sum_{t=0}^{E-1}\boldsymbol{r}_{t,k} + \frac{1}{2}\sum_{t=1}^{E-1}\left(\boldsymbol{\omega} + \boldsymbol{M}_{t,k}\boldsymbol{\varepsilon}_k\right)^{\mathrm{T}}, \tag{12}$$

where $\boldsymbol{M}_{t,k}$ is a projection matrix onto the range space of $\boldsymbol{\Phi}_t$ under the metric $\boldsymbol{J}^{-1}$, and $\boldsymbol{r}_{t,k}$ is the immediate cost of the k^{th} trajectory at time t.

For each k^{th} roll-out trajectory, the immediate cost function calculated from the sensing signal feedback is defined as follows:

$$r_t = \left(q_t - q_t^{\mathrm{d}}\right)^2, \tag{13}$$

where q_t represents the joint angle of the exoskeleton, and q_t^{d} the desired position of the wearer. Then the overall trajectory cost R is

$$R = \sqrt{\frac{1}{E}\sum_{t=1}^{E} r_t}. \tag{14}$$

The probability of the k^{th} roll-out trajectory is obtained by mapping the cost of each trajectory to [0, 1] through the softmax function, as shown in Eq. (15):

$$\boldsymbol{P}_k = \mathrm{e}^{-\frac{1}{\gamma}\boldsymbol{S}_k} \Big/ \sum_{k=1}^{K} \mathrm{e}^{-\frac{1}{\gamma}\boldsymbol{S}_k}, \tag{15}$$

where the parameter γ is a constant coefficient within (0, 1]. It can be seen from Eq. (15) that the higher the cost, the lower the probability, thus ensuring $\mathrm{PI}^{\mathrm{BB}}$ to converge to a value with low cost.

The final parameter is updated through reward-weighted averaging:

$$\delta\boldsymbol{\omega} = \boldsymbol{P}_k \cdot \boldsymbol{\varepsilon}_k . \tag{16}$$

The process of $\mathrm{PI}^{\mathrm{BB}}$ for motion model adaptation is shown in Algorithm 1, which corresponds to the clear display of the dashed box shown in Fig. 1. The index notations in this paper are listed in Table 1.

Algorithm 1 Motion adaptation

Input: initial state of the parameter $\boldsymbol{\omega}$ ($\delta\boldsymbol{\omega}$ is weighted averaging), the basis function $\boldsymbol{\Phi}_t$, and desire trajectory $\boldsymbol{y}^{\mathrm{d}}$

Output: parameter vector $\boldsymbol{\omega}$

for k =1, 2, ⋯, K

  Sample $\boldsymbol{\varepsilon}_k \sim \mathcal{N}(0, \boldsymbol{\Sigma})$

  Roll-out: $\boldsymbol{y}_{t,k} = \boldsymbol{\Phi}_t^{\mathrm{T}}(\boldsymbol{\omega} + \boldsymbol{\varepsilon}_k)$

  $$\boldsymbol{r}_{t,k} = \left(\boldsymbol{q}_{t,k} - \boldsymbol{q}_{t,k}^{\mathrm{d}}\right)^2$$

  Compute trajectory cost: $\boldsymbol{M}_{t,k} = \dfrac{\boldsymbol{J}^{-1}\boldsymbol{\Phi}_{t,k}\boldsymbol{\Phi}_{t,k}^{\mathrm{T}}}{\boldsymbol{\Phi}_{t,k}^{\mathrm{T}}\boldsymbol{J}^{-1}\boldsymbol{\Phi}_{t,k}}$

  $$\boldsymbol{S}_k = \sum_{t=0}^{E-1} \boldsymbol{r}_{t,k} + \frac{1}{2}\sum_{t=1}^{E-1}\left(\boldsymbol{\omega} + \boldsymbol{M}_{t,k}\boldsymbol{\varepsilon}_k\right)^{\mathrm{T}}$$

end for

for k =1, 2, ⋯, K

  Compute the probability of each roll-out:

  $$\boldsymbol{P}_k = \mathrm{e}^{-\frac{1}{\gamma}\boldsymbol{S}_k} \Big/ \sum_{k=1}^{K} \mathrm{e}^{-\frac{1}{\gamma}\boldsymbol{S}_k}$$

end for

Cost-weighted averaging: $\delta\boldsymbol{\omega} = \boldsymbol{P}_k \cdot \boldsymbol{\varepsilon}_k$

Update: $\boldsymbol{\omega}^{\mathrm{new}} \leftarrow \boldsymbol{\omega} + \delta\boldsymbol{\omega}$

Overall trajectory cost: $R = \sqrt{\dfrac{1}{E}\sum_{t=1}^{E} r_t}$

until the overall trajectory R cost converges

Table 1 Definition of the index notations

Index	Definition
i	Number of basis functions
N	Maximum number of basis functions
t	Number of time steps
E	Maximum number of time steps
k	Number of roll-out trajectories
K	Maximum number of roll-out trajectories
m	Number of demonstration trajectories
M	Maximum number of demonstration trajectories

3 Simulations and experiments

In this section, we describe simulations and experiments conducted on a lower limb exoskeleton to verify the proposed motion learning scheme. To test the feasibility of the proposed method before implementing it on the hardware platform, we performed simulations.

3.1 Simulations

3.1.1 Motion generation

Simulations were implemented to first verify the representation ability of the proposed motion model, and compare it with the existing classical methods. First, a curve generated by the second-order Fourier series was used to imitate a human walking trajectory:

$$p_{\mathrm{d}} = -\cos(2\pi t) + \sin(2\pi t) + \cos(4\pi t) + \sin(4\pi t). \tag{17}$$

This reference trajectory consists of periodic sine waves with different frequencies and amplitudes. ProMP represents and learns the demonstration trajectory based on Section 2.2. The regression parameter λ is generally set to 0.01, and the basis function width h is 0.05. The number of basis functions is crucial to the representative ability of primitives. Fig. 2 shows the trajectories learned from different numbers N of basis functions, and Fig. 2b is the root mean square error (RMSE) between the learned trajectory and the target trajectory. The representation ability is weak when N is small, but grows extremely fast as N increases. The trajectory learned by 10 basis functions, shown by the brown line in Fig. 2a, coincides exactly with the target, as shown by the dashed blue line, and the RMSE is within 0.003.

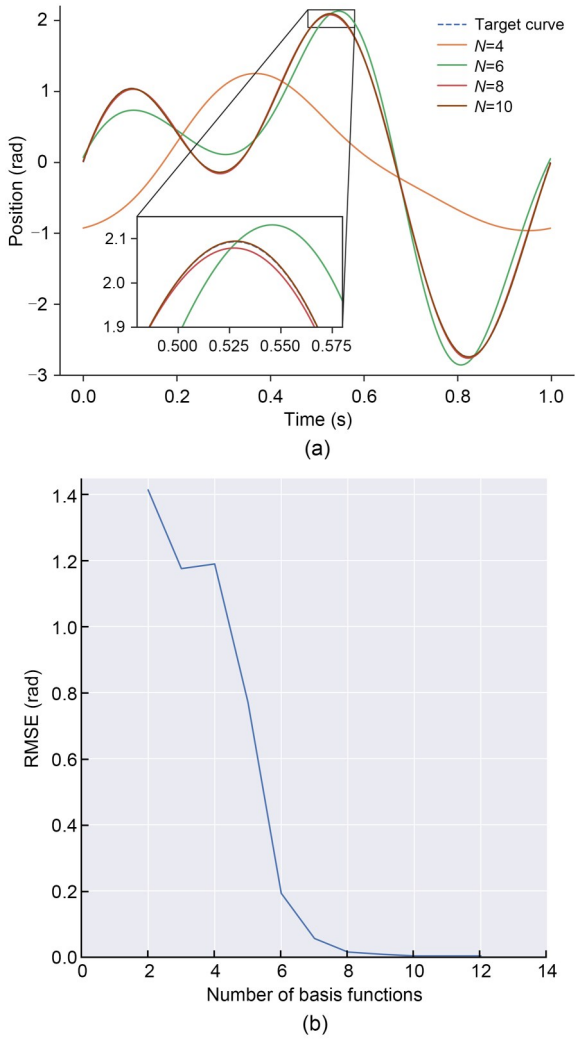

Fig. 2 Trajectory learning by ProMP under basis functions of different numbers: (a) learning curve; (b) learning cost
References to color refer to the online version of this figure

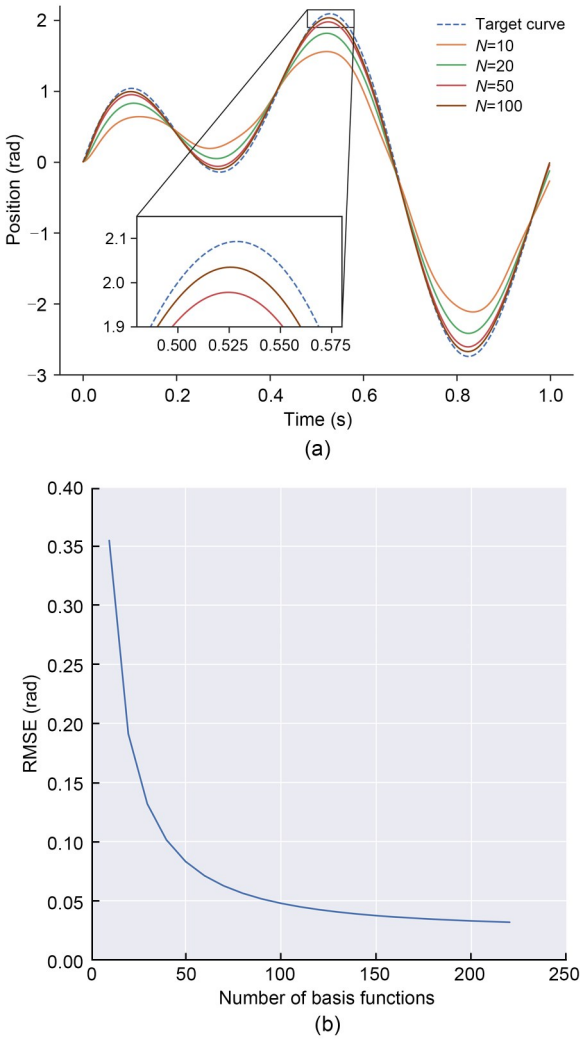

Fig. 3 Trajectory learning by DMP under basis functions of different numbers: (a) learning curve; (b) learning cost
References to color refer to the online version of this figure

To reveal the representation ability of ProMP in this case, the commonly used trajectory representation DMP was also adopted to learn the reference trajectory for comparison. The performance of DMP learning under different numbers of basis functions is shown in Fig. 3.

For DMP, the RMSE was not as large as that of ProMP at the beginning, but as the number of functions increased, the improvement in RMSE was very small. Even with 10 basis functions when the ProMP completely converged, the trajectory of DMP was far from the target trajectory. The final convergence curve was still clearly separate from the target trajectory, with the final RMSE being around 0.03 rad, which is 10 times that of ProMP. Using LWR to learn the weights $\boldsymbol{\omega}$ of DMP, it was almost impossible to achieve the same RMSE as ProMP for this kind of trajectory based on our simulation. To some extent, the smaller number of basis functions needed indicated the stronger representative ability of the approach. Besides, the number of basis functions was proportional to the computational consumption, so fewer basis functions are friendly to the real-time effect of the strategy. Therefore, ProMP achieved a better performance by representing a trajectory with great accuracy and efficiency. In addition, the superior performance of ProMP over DMP was confirmed in a study involving stroke-based movements (Paraschos et al., 2013).

3.1.2 Trajectory adaptation

For online motion generation of the exoskeleton, powerful representation ability is essential, but not sufficient. The representative tool must be adaptive and can reproduce a new trajectory precisely as soon as possible. We conducted a simulation to demonstrate the online adaptation and updating of the proposed method, ProMP combined with PI^{BB} (ProMP-PI^{BB}). The performance of the baseline DMP combined with PI^2 (DMP-PI^2) in the same situation was also evaluated. The adopted number of basis functions of ProMP was 10 according to Section 3.1.1. To ensure that the initial trajectories of DMP and ProMP were as similar as possible, the basis function number of DMP needed to be 150. It was assumed that Eq. (17) was the current trajectory, and that the target trajectory was similar, but had different frequencies and amplitudes:

$$p_t = -0.5\cos(2\pi t) + \sin(2\pi t) + 0.5\cos(4\pi t) + \sin(4\pi t). \tag{18}$$

The initial value of $\boldsymbol{\omega}$ corresponded to the current trajectory, and was updated every gait cycle. The time steps were normalized to 150 based on the time interval of each gait cycle. During learning, $k = 50$ roll-outs were performed for one update. Fig. 4 shows the trajectory adaptation processes of ProMP-PI^{BB} and DMP-PI^2. For brevity, only a few representative time nodes are shown. The trajectory updated by ProMP-PI^{BB} was very close to the target trajectory at only the 5th update, while the DMP-PI^2 trajectory was still close to the beginning. ProMP-PI^{BB}'s trajectory adapted to the target perfectly from the 15th update. The fitting process of DMP-PI^2 kept a constant speed. In the end, 20 updates were needed to achieve convergence to the target, and the final convergence performance still had visible misfits compared to that of ProMP-PI^{BB}.

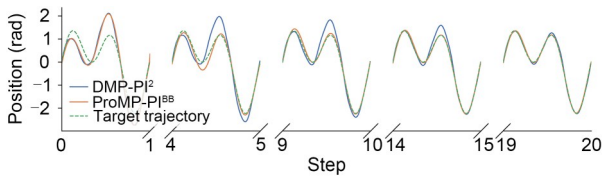

Fig. 4 Trajectory adaptation processes of ProMP-PI^{BB} and DMP-PI^2

References to color refer to the online version of this figure

To fully evaluate the adaptation efficiency, the learning costs are shown in Fig. 5. The convergence speed of ProMP-PI^{BB} was distinctly higher than that of DMP-PI^2. The trajectory of ProMP-PI^{BB} took only about 10 updates to converge. However, DMP-PI^2 took at least 20 updates to reach the lower cost, and the final cost was also worse than that of ProMP-PI^{BB}. This confirmed that ProMP-PI^{BB} outperforms DMP-PI^2 in terms of convergence speed and final cost. Furthermore, DMP-PI^2 had a disadvantage in terms of computation time per update. First, the programming logic (i.e., the complexity of the code) of ProMP is simple, so it reduces the computational complexity. Second, DMP needs many more basis functions to achieve the same performance, which will cost more time. Therefore, in principle, the calculation time of DMP for each update is much longer than that of ProMP. The situation was the same for PI^{BB} because it is essentially a simplification of PI^2. In the simulation, the ProMP-PI^{BB} update time was about 0.46 s with 10 basis functions. With DMP-PI^2, the update time was 1.86 s for 10 basis functions, and 10.14 s for 150 basis functions.

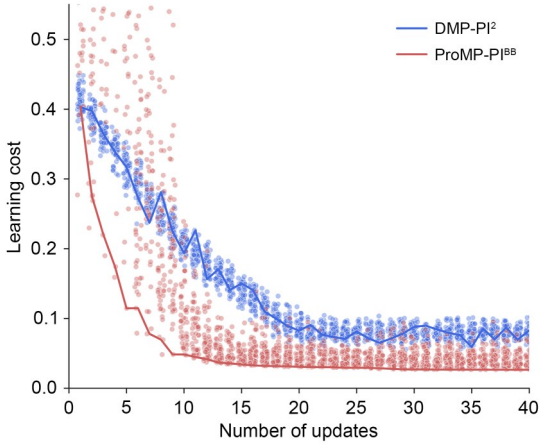

Fig. 5 Learning costs during trajectory adaptation of ProMP-PI^{BB} and DMP-PI^2

References to color refer to the online version of this figure

3.2 Experiments on lower limb exoskeleton

3.2.1 Hardware

In this subsection, we describe experiments implemented on a lower limb exoskeleton system, HEXO, developed in our lab. HEXO is an anthropomorphic device, which has similar DOFs to the human lower

limb. Fig. 6 shows the main components of HEXO. The backpack was equipped with an Advanced RISC Machines (ARM) control panel, power supply, and data acquisition card. There were four active DOFs for hip and knee flexion/extension. The actuation system was powered by a brushless DC motor. An incremental encoder was integrated into the motor. The motor was combined with a harmonic drive with a ratio of 1:100 in the hip joint, and 1:80 in the knee joint. Lower limb motion was measured by an inertial measurement unit (IMU). Torque sensors were placed at joints. Three six-axis force sensors (SFSs) were installed at the back, and sensing-shoes between the wearer and the exoskeleton were to perceive the human-robot interaction force. All sensor data were transmitted to the ARM panel through a controller area network (CAN) bus, whose transmission rate was up to 1 Mb/s.

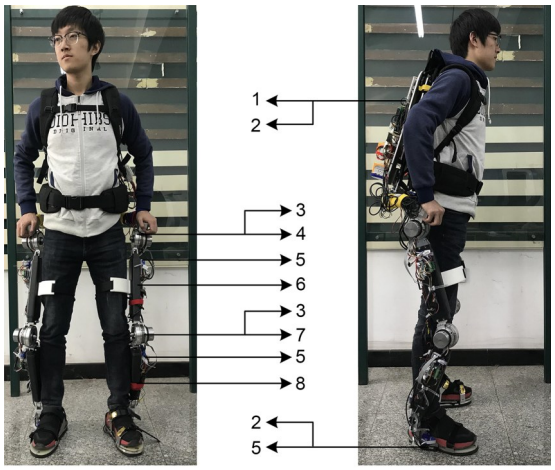

Fig. 6 The hardware system of HEXO exoskeleton
1, backpack; 2, six-axis force sensors; 3, encode and torque sensor; 4, hip joint; 5, inertial measurement unit; 6, thigh limb; 7, knee joint; 8, calf limb

3.2.2 Experimental protocol

As shown in Fig. 1, the motion trajectory first needed to be learned offline before the online experiment was carried out on HEXO. Three voluntary subjects 1, 2, and 3, whose characteristics are listed in Table 2, participated in the data acquisition. Subjects 1, 2, and 3 were asked to perform level walking on a treadmill at their normal speed. The exoskeleton HEXO was working in zero-force mode with no enabled torque assistance, to obtain the most natural gait of the subjects when wearing the exoskeleton.

Table 2 Detailed information of the six subjects

Subject index	Gender	Age (year)	Height (cm)	Weight (kg)
1	Male	24	172	66
2	Male	23	180	61
3	Male	24	176	72.5
4	Male	23	187	70
5	Female	24	169	58.5
6	Male	45	179	78

In both offline and online experiments, trajectories of all four joints of the HEXO were generated simultaneously. Only the left leg data are shown in all figures of this paper, because the properties of the two legs are similar.

3.2.3 Motion initial learning

The trajectory data were first obtained, and the next step was to represent and learn the trajectory using ProMP. The simulation in Section 3.1.1 verified that ProMP is a powerful representative tool, but another beneficial property of ProMP is that it can concurrently activate multiple primitives, i.e., learning multiple trajectories. Fig. 7 shows the cut and normalized walking trajectories of subjects 1, 2, and 3 according to the gait cycle. Figs. 7a and 7b show the mean and covariance of the hip and knee data, respectively. The general trend of the curve was the same for each joint, but the shape of the curve differed, even when the subjects had similar heights and weights. The red areas of Figs. 7c and 7d showed the trajectories learned by ProMP from all three subjects, and contained all the possibilities. The red line can be regarded as the average of all acquired trajectories, so it is more representative than any others. Besides, the more trajectories learned, the more general the reference trajectory.

3.2.4 Motion online adaptation

The trajectory learned offline is regarded as the reference trajectory when working online. The exoskeleton took several gait cycles to learn the optimal parameters $\boldsymbol{\omega}$ based on the initial one, which made the trajectory cost R converge. The experiment was implemented to test the effect of online adaptation of the proposed method. We included baseline DMP-PI^2 for comparison. Previous results could not be used

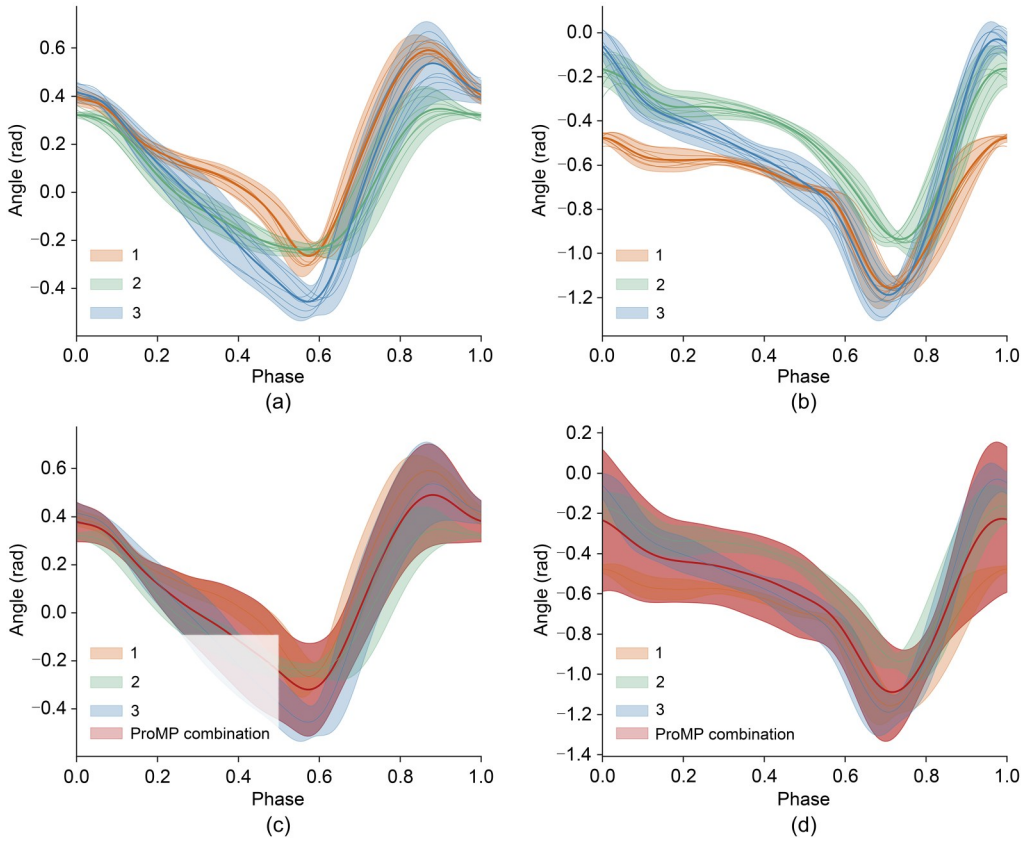

Fig. 7 Motion initial learning by ProMP from subjects 1, 2, and 3: (a) mean and covariance of the hip trajectory data; (b) mean and covariance of the knee trajectory data; (c) hip trajectory learned by ProMP; (d) knee trajectory learned by ProMP
References to color refer to the online version of this figure

for comparison because of different criteria and experimental conditions, so we reproduced the DMP-PI^2 on our platform with the same conditions as in our proposed ProMP-PI^{BB}. To avoid too much calculation time, 20 commonly used basis functions were selected for DMP.

In this experiment, there were three new subjects, 4, 5, and 6, as listed in Table 2. To evaluate the effectiveness of the method, subjects with differences were deliberately selected for validation. For example, subject 5 was a female with a lower height and subject 6 was much older than other subjects. The subjects were also asked to deliberately change their speeds several times when performing level walking, to test the adaptability of the method to different speeds.

Fig. 8 shows the online trajectories generated by ProMP-PI^{BB} and DMP-PI^2, and the actual trajectory for subject 4, that is, the process of motion adaptation. Comparing ProMP-PI^{BB} with DMP-PI^2, the initial error of the first step before learning was almost the same. However, the trajectory generated by ProMP-PI^{BB} converged to the desired trajectory after about the fourth step for both the knee and hip joints. DMP-PI^2 did not converge until the sixth step for the hip and the seventh for the knee, and the fitting of the trajectory was not good for the hip.

Motion adaptation also includes temporal modulation. Temporal modulation is a valuable property as it enables the motion model to be applied to walking when the speed changes. After all, it is inevitable that speed changes during human walking. The speed of the first two steps was very stable and the curve fitted well (Fig. 9). The speed was slightly lower from the third step and could be adjusted immediately. The gait changed slightly in the fifth step, the stride became larger, and the next step was adjusted to this quickly, as shown by the red curve. Starting from the seventh step, the speed increased, and the generated trajectory was constantly adjusted. When the speed started to

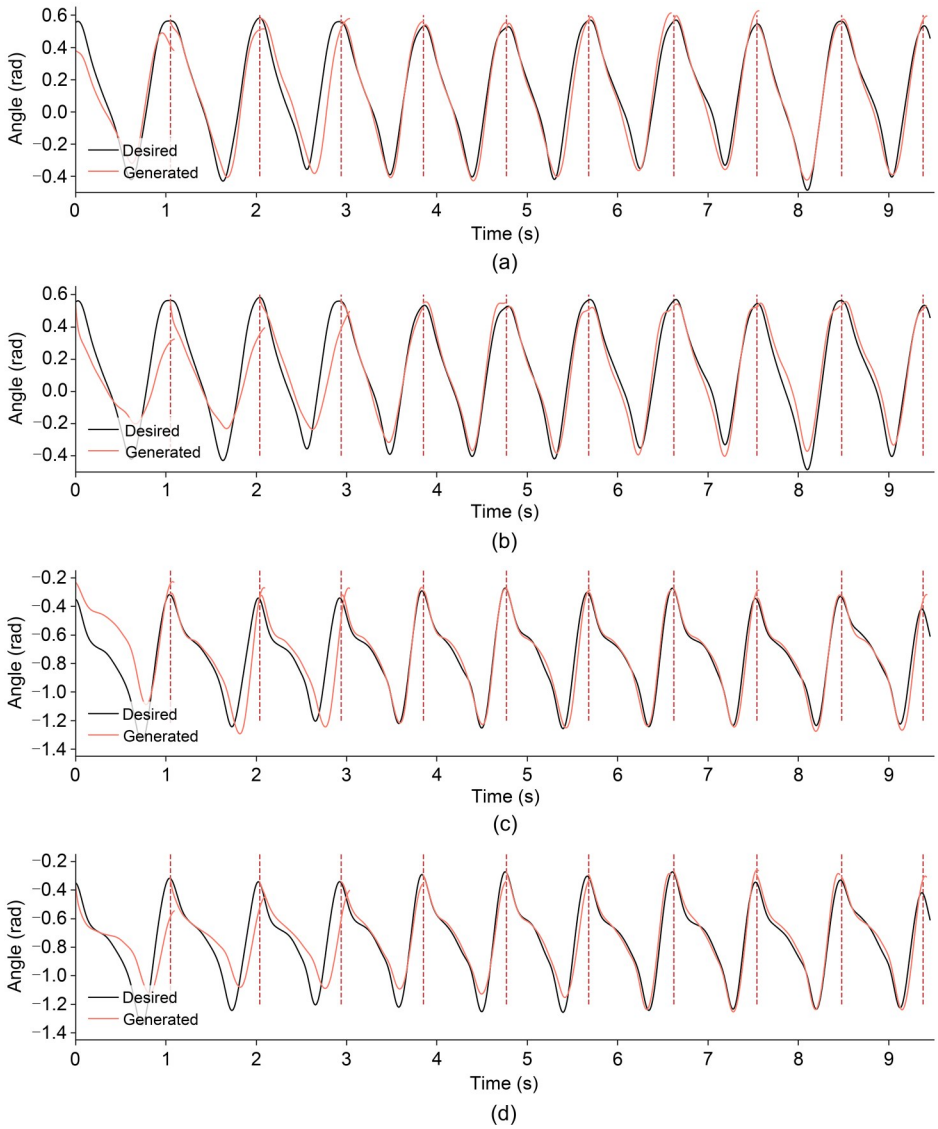

Fig. 8 Motion adaptation process of ProMP-PI^{BB} and DMP-PI^2: (a) hip of ProMP-PI^{BB}; (b) hip of DMP-PI^2; (c) knee of ProMP-PI^{BB}; (d) knee of DMP-PI^2

References to color refer to the online version of this figure

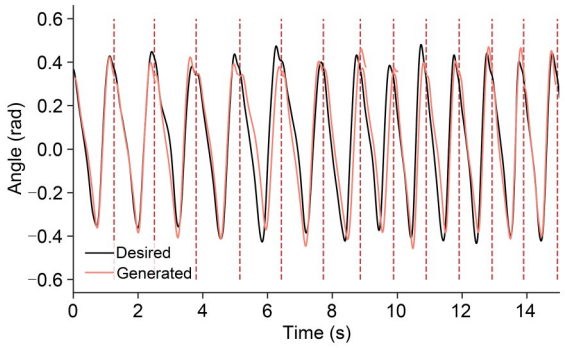

Fig. 9 Trajectory adaptation performance of the proposed ProMP-PI^{BB} when the walking speed changes

References to color refer to the online version of this figure

stabilize after the ninth step, the trajectory was almost stable. Therefore, the trajectory can be adapted quickly when gait or walking speed changes.

Table 3 summarizes the experiment results of the three subjects for DMP-PI^2 and ProMP-PI^{BB}. It shows the RMSEs before adaptation (the first step) and after adaptation, the convergence step, and the improvement rate of the proposed method. Before updating, the RMSEs of ProMP were lower than those of DMP. In the end of the adaptation, the final RMSEs of ProMP-PI^{BB} were also smaller than those of DMP-PI^2. The average improvement for the three

Table 3 The adaptation experiment results (hip/knee) of ProMP-PI^{BB} and DMP-PI^2 for three subjects

Subject index	RMSE (rad)				Convergence step	
	Before adaptation		After adaptation			
	DMP	ProMP	DMP-PI^2	ProMP-PI^{BB}	DMP-PI^2	ProMP-PI^{BB}
4	0.182/0.195	0.101/0.240	0.052/0.040	0.048/0.039	5th/7th	4th/4th
5	0.301/0.413	0.198/0.288	0.082/0.099	0.065/0.089	7th/9th	3rd/5th
6	0.087/0.156	0.076/0.123	0.073/0.079	0.057/0.061	6th/8th	3rd/3rd

subjects was 15.49%, indicating that the proposed strategy achieved better performance. Although the final errors of the two methods were both small, a little mismatch between the desired trajectory and the generated trajectory will cause huge human-robot interaction resistance when the exoskeleton was working. Therefore, any improvement that can reduce the error is valuable. In addition, for ProMP and ProMP-PI^{BB}, the error convergence was not so obvious. This was mainly because the initial reference trajectory learned by ProMP was already a general trajectory, so there was no need for much adjustment during trajectory adaptation. Besides, every step of a person in the walking process cannot be exactly the same, so the error of the generated trajectory must fluctuate, even after convergence. Fig. 8 shows that the converged trajectory was very similar to the desired trajectory, but there were still slight mismatches.

Most importantly, the convergence time was the same for all experimental subjects. With ProMP-PI^{BB} the trajectory converged at around the fourth gait cycle, but DMP-PI^2 needed seven or more cycles. Whether for the hip joint or knee joint, the proposed method could generate a trajectory suitable for the current wearer in only three or four steps. Furthermore, the trajectory generation errors of the knee joint were a bit larger and more unstable than those of the hip joint, because the movement of the knee joint is more complicated.

The effect of trajectory generation was also affected by the characteristics of the subjects. Among the three subjects, 4, 5, and 6, the performance of subject 5 was the worst. The reason may be that subject 5 was a female who had the smallest height and weight, very different from the three subjects learning offline, 1, 2, and 3. The initial error of the first step of subject 6 was the lowest, which is reasonable as his physical characteristics were the closest to those of the three subjects. However, the error of subject 6 was the most unstable, perhaps because of his unstable gait.

The experiment results showed that our proposed method features faster convergence and a smaller cost compared with the baseline DMP-PI^2. Moreover, as stated in Section 3.1, ProMP-PI^{BB} has a much lower calculation consumption for each update. Above all, the proposed motion learning scheme is a reliable high-level approach for exoskeleton control. It generates trajectories in real time, in parallel with human intention, and can quickly react to different subjects and variable situations.

4 Conclusions and future work

In this paper, we propose a novel motion learning scheme to generate a motion trajectory online for lower limb exoskeletons. There are two complementary aspects of this novel scheme: motion generation and motion adaptation. For motion generation, the motion is modeled by ProMP with offline initial learning using pre-collected trajectories. For motion adaptation, the motion model based on ProMP can be further learned and updated online using the black-box optimization PI^{BB}. This is the first time that ProMP has been adopted to model motion for an exoskeleton. The simulation and experiment results showed that this motion learning can generate a trajectory online in parallel with human intention quickly and accurately, and most importantly, the learning speed is much higher than those of the existing methods. The experiments verified that the proposed strategy has a better performance than the existing popular strategies, not only with a higher convergence rate, but also a lower final cost. Therefore, the exoskeleton with the proposed motion learning is able to adapt to different wearers and variable environments in a timely manner. This

human-exoskeleton system can co-work collaboratively faster and more consistently, and with a better human-robot interaction. The combination of ProMP and PI^{BB} produces an even better effect.

In the future, the motion learning will be tested under assistance mode to complete the exoskeleton function. The appropriate control method and corresponding results will be analyzed in detail. For preliminary testing of the effect of the proposed method, the motion modes tested in this study involved only ground-level walking. In the future, all basic rhythmic locomotion modes in daily living will be included, such as stair ascent, stair descent, ramp ascent, and ramp descent.

Contributors

Jiaqi WANG conducted the research and drafted the paper. Yongzhuo GAO, Dongmei WU, and Wei DONG revised and finalized the paper.

Compliance with ethics guidelines

Jiaqi WANG, Yongzhuo GAO, Dongmei WU, and Wei DONG declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Al-Shuka HFN, Corves B, Zhu WH, et al., 2016. Multi-level control of zero-moment point-based humanoid biped robots: a review. *Robotica*, 34(11):2440-2466. https://doi.org/10.1017/S0263574715000107

Colombo G, Joerg M, Schreier R, et al., 2000. Treadmill training of paraplegic patients using a robotic orthosis. *J Rehabil Res Dev*, 37(6):693-700.

d'Avella A, Bizzi E, 2005. Shared and specific muscle synergies in natural motor behaviors. *Proc Nat Acad Sci USA*, 102(8):3076-3081. https://doi.org/10.1073/pnas.0500199102

Deng MD, Li ZJ, Kang Y, et al., 2020. A learning-based hierarchical control scheme for an exoskeleton robot in human-robot cooperative manipulation. *IEEE Trans Cybern*, 50(1): 112-125. https://doi.org/10.1109/TCYB.2018.2864784

Esquenazi A, Talaty M, Packel A, et al., 2012. The ReWalk powered exoskeleton to restore ambulatory function to individuals with thoracic-level motor-complete spinal cord injury. *Am J Phys Med Rehabil*, 91(11):911-921. https://doi.org/10.1097/PHM.0b013e318269d9a3

Fu CL, Chen K, 2008. Gait synthesis and sensory control of stair climbing for a humanoid robot. *IEEE Trans Ind Electron*, 55(5):2111-2120. https://doi.org/10.1109/TIE.2008.921205

Guizzo E, Goldstein H, 2005. The rise of the body bots [robotic exoskeletons]. *IEEE Spectr*, 42(10):50-56. https://doi.org/10.1109/MSPEC.2005.1515961

Hassan M, Kadone H, Suzuki K, et al., 2014. Wearable gait measurement system with an instrumented cane for exoskeleton control. *Sensors*, 14(1):1705-1722. https://doi.org/10.3390/s140101705

He W, Li ZJ, Chen CLP, 2017. A survey of human-centered intelligent robots: issues and challenges. *IEEE/CAA J Autom Sin*, 4(4):602-609. https://doi.org/10.1109/JAS.2017.7510604

Huang R, Cheng H, Guo H, et al., 2018. Hierarchical learning control with physical human-exoskeleton interaction. *Inform Sci*, 432:584-595. https://doi.org/10.1016/j.ins.2017.09.068

Huang R, Cheng H, Qiu J, et al., 2019. Learning physical human-robot interaction with coupled cooperative primitives for a lower exoskeleton. *IEEE Trans Autom Sci Eng*, 16(4): 1566-1574. https://doi.org/10.1109/TASE.2018.2886376

Ijspeert AJ, Nakanishi J, Schaal S, 2002. Movement imitation with nonlinear dynamical systems in humanoid robots. Proc IEEE Int Conf on Robotics and Automation, p.1398-1403. https://doi.org/10.1109/ROBOT.2002.1014739

Ijspeert AJ, Nakanishi J, Hoffmann H, et al., 2013. Dynamical movement primitives: learning attractor models for motor behaviors. *Neur Comput*, 25(2):328-373. https://doi.org/10.1162/NECO_a_00393

Jenison RL, Fissell K, 1995. A comparison of the von Mises and Gaussian basis functions for approximating spherical acoustic scatter. *IEEE Trans Neur Netw*, 6(5): 1284-1287. https://doi.org/10.1109/72.410375

Kagawa T, Ishikawa H, Kato T, et al., 2015. Optimization-based motion planning in joint space for walking assistance with wearable robot. *IEEE Trans Rob*, 31(2):415-424. https://doi.org/10.1109/TRO.2015.2409434

Kazemi J, Ozgoli S, 2019. Real-time walking pattern generation for a lower limb exoskeleton, implemented on the exoped robot. *Rob Auton Syst*, 116:1-23. https://doi.org/10.1016/j.robot.2019.02.012

Kazerooni H, Steger R, 2006. The Berkeley lower extremity exoskeleton. *J Dynam Syst Meas Contr*, 128(1):14-25. https://doi.org/10.1115/1.2168164

Komura T, Nagano A, Leung H, et al., 2005. Simulating pathological gait using the enhanced linear inverted pendulum model. *IEEE Trans Biomed Eng*, 52(9):1502-1513. https://doi.org/10.1109/TBME.2005.851530

Krüger V, Kragic D, Ude A, et al., 2007. The meaning of action: a review on action recognition and mapping. *Adv Rob*, 21(13):1473-1501. https://doi.org/10.1163/156855307782148578

Kulić D, Ott C, Lee D, et al., 2012. Incremental learning of full body motion primitives and their sequencing through human motion observation. *Int J Rob Res*, 31(3):330-345. https://doi.org/10.1177/0278364911426178

Lee SW, Yi T, Jung JW, et al., 2015. Design of a gait phase recognition system that can cope with EMG electrode location variation. *IEEE Trans Autom Sci Eng*, 14(3):1429-1439. https://doi.org/10.1109/TASE.2015.2477283

Paraschos A, Daniel C, Peters J, et al., 2013. Probabilistic movement primitives. Proc 26th Int Conf on Neural Information Processing Systems, p.2616-2624.

Paraschos A, Daniel C, Peters J, et al., 2018. Using probabilistic movement primitives in robotics. *Auton Rob*, 42(3): 529-551. https://doi.org/10.1007/s10514-017-9648-7

Sankai Y, 2010. HAL: hybrid assistive limb based on cybernics. In: Kaneko M, Nakamura Y (Eds.), Robotics Research. Springer, Berlin, Heidelberg, p.25-34. https://doi.org/10.1007/978-3-642-14743-2_3

Sanz-Merodio D, Cestari M, Arevalo JC, et al., 2014. Generation and control of adaptive gaits in lower-limb exoskeletons for motion assistance. *Adv Rob*, 28(5):329-338. https://doi.org/10.1080/01691864.2013.867284

Schaal S, Ijspeert A, Billard A, 2003. Computational approaches to motor learning by imitation. *Phil Trans R Soc B Biol Sci*, 358(1431):537-547. https://doi.org/10.1098/rstb.2002.1258

Schaal S, Peters J, Nakanishi J, et al., 2005. Learning movement primitives. 11th Int Symp on Robotics Research, p.561-572. https://doi.org/10.1007/11008941_60

Schmidhuber J, 2015. Deep learning in neural networks: an overview. *Neur Netw*, 61:85-117. https://doi.org/10.1016/j.neunet.2014.09.003

Spiegelhalter DJ, Best NG, Carlin BP, et al., 2002. Bayesian measures of model complexity and fit. *J R Stat Soc Ser B* (*Stat Methodol*), 64(4):583-639. https://doi.org/10.1111/1467-9868.00353

Strausser KA, Kazerooni H, 2011. The development and testing of a human machine interface for a mobile medical exoskeleton. Proc IEEE/RSJ Int Conf on Intelligent Robots and Systems, p.4911-4916. https://doi.org/10.1109/IROS.2011.6095025

Stulp F, Sigaud O, 2012. Policy Improvement Methods: Between Black-Box Optimization and Episodic Reinforcement Learning. https://hal.archives-ouvertes.fr/hal-00738463 [Accessed on Jan. 12, 2021]

Theodorou E, Buchli J, Schaal S, 2010. A generalized path integral control approach to reinforcement learning. *J Mach Learn Res*, 11:3137-3181.

Todorov E, Jordan MI, 2002. Optimal feedback control as a theory of motor coordination. *Nat Neurosci*, 5(11):1226-1235. https://doi.org/10.1038/nn963

Tucker MR, Olivier J, Pagel A, et al., 2015. Control strategies for active lower extremity prosthetics and orthotics: a review. *J Neuroeng Rehabil*, 12(1):1. https://doi.org/10.1186/1743-0003-12-1

Veneman JF, Kruidhof R, Hekman EEG, et al., 2007. Design and evaluation of the LOPES exoskeleton robot for interactive gait rehabilitation. *IEEE Trans Neur Syst Rehabil Eng*, 15(3):379-386. https://doi.org/10.1109/TNSRE.2007.903919

Vukobratović M, Borovac B, 2004. Zero-moment point—thirty five years of its life. *Int J Human Rob*, 1(1):157-173. https://doi.org/10.1142/S0219843604000083

Walsh CJ, Endo K, Herr H, 2007. A quasi-passive leg exoskeleton for load-carrying augmentation. *Int J Human Rob*, 4(3): 487-506. https://doi.org/10.1142/S0219843607001126

Xu B, Sun FC, 2018. Composite intelligent learning control of strict-feedback systems with disturbance. *IEEE Trans Cybern*, 48(2):730-741. https://doi.org/10.1109/TCYB.2017.2655053

Yan TF, Cempini M, Oddo CM, et al., 2015. Review of assistive strategies in powered lower-limb orthoses and exoskeletons. *Rob Auton Syst*, 64:120-136. https://doi.org/10.1016/j.robot.2014.09.032

Yang CG, Chen CZ, Wang N, et al., 2019. Biologically inspired motion modeling and neural control for robot learning from demonstrations. *IEEE Trans Cogn Dev Syst*, 11(2): 281-291. https://doi.org/10.1109/TCDS.2018.2866477

Yuan YX, Li ZJ, Zhao T, et al., 2020. DMP-based motion generation for a walking exoskeleton robot using reinforcement learning. *IEEE Trans Ind Electron*, 67(5):3830-3839. https://doi.org/10.1109/TIE.2019.2916396

Zoss AB, Kazerooni H, Chu A, 2006. Biomechanical design of the Berkeley lower extremity exoskeleton (BLEEX). *IEEE/ASME Trans Mechatr*, 11(2):128-138. https://doi.org/10.1109/TMECH.2006.871087