Reinforcement Learning and Control of a Lower Extremity Exoskeleton for Squat Assistance

Shuzhen Luo*[1], *Ghaith Androwis*[1,2], *Sergei Adamovich*[1], *Hao Su*[3], *Erick Nunez*[1] and *Xianlian Zhou*[1]

[1]Department of Biomedical Engineering, New Jersey Institute of Technology, Newark, NJ, United States, [2]Kessler Foundation, West Orange, Newark, NJ, United States, [3]Department of Mechanical Engineering, The City University of New York, Newark, NY, United States

OPEN ACCESS

Edited by:
Nitin Sharma,
North Carolina State University,
United States

Reviewed by:
Yue Chen,
University of Arkansas, United States
Xiao Xiao,
Southern University of Science and Technology, China

***Correspondence:**
Xianlian Zhou
alexzhou@njit.edu

Specialty section:
This article was submitted to Biomedical Robotics, a section of the journal Frontiers in Robotics and AI

Received: *30 April 2021*
Accepted: *29 June 2021*
Published: *19 July 2021*

Citation:
Luo S, Androwis G, Adamovich S, Su H, Nunez E and Zhou X (2021) Reinforcement Learning and Control of a Lower Extremity Exoskeleton for Squat Assistance. Front. Robot. AI 8:702845. doi: 10.3389/frobt.2021.702845

A significant challenge for the control of a robotic lower extremity rehabilitation exoskeleton is to ensure stability and robustness during programmed tasks or motions, which is crucial for the safety of the mobility-impaired user. Due to various levels of the user's disability, the human-exoskeleton interaction forces and external perturbations are unpredictable and could vary substantially and cause conventional motion controllers to behave unreliably or the robot to fall down. In this work, we propose a new, reinforcement learning-based, motion controller for a lower extremity rehabilitation exoskeleton, aiming to perform collaborative squatting exercises with efficiency, stability, and strong robustness. Unlike most existing rehabilitation exoskeletons, our exoskeleton has ankle actuation on both sagittal and front planes and is equipped with multiple foot force sensors to estimate center of pressure (CoP), an important indicator of system balance. This proposed motion controller takes advantage of the CoP information by incorporating it in the state input of the control policy network and adding it to the reward during the learning to maintain a well balanced system state during motions. In addition, we use dynamics randomization and adversary force perturbations including large human interaction forces during the training to further improve control robustness. To evaluate the effectiveness of the learning controller, we conduct numerical experiments with different settings to demonstrate its remarkable ability on controlling the exoskeleton to repetitively perform well balanced and robust squatting motions under strong perturbations and realistic human interaction forces.

Keywords: lower extremity rehabilitation exoskeleton, reinforcement learning, center of pressure, balanced squatting control, human-exoskeleton interaction

1 INTRODUCTION

Due to the aging population and other factors, an increasing number of people are suffering from neurological disorders, such as stroke, central nervous system disorder, and spinal cord injury (SCI) that affect the patient's mobility. Emerging from the field of robotics, robotic exoskeletons have become a promising solution to enable mobility impaired people to perform the activities of daily living (ADLs) (Mergner and Lippi, 2018; Vouga et al., 2017; Zhang et al., 2018). Lower-limb rehabilitation exoskeletons are wearable bionic devices that are equipped with powerful actuators to assist people to regain their lower leg function and mobility. With a built-in multi-sensor system, an exoskeleton can recognise the wearer's motion intentions and assist the wearer's motion accordingly (Chen et al., 2016). Compared to traditional physical therapy, rehabilitation exoskeleton robots have the advantages of providing more intensive patient repetitive training, better quantitative feedback, and improved life quality for patients (Chen et al., 2016).

A degradation or loss of balance as a result of neuromusculoskeletal disorders or impairments is a common symptom, for instance, in patients with SCI or stroke. Balance training in the presence of external perturbations (Horak et al., 1997) is considered as one of the more important factors in evaluating patients' rehabilitation performance. A rehabilitation exoskeleton can be employed for balance training to achieve static stability (quiet standing) or dynamic stability (squatting, sit-to-stand, and walking) (Bayon et al., 2020; Mungai and Grizzle, 2020; Rajasekaran et al., 2015). Squatting exercises are very common for resistance-training programs because their multiple-joint movements are a characteristic of most sports and daily living activities. In rehabilitation, squatting is commonly performed as an important exercise for patients during the recovery of various lower extremity injuries (McGinty et al., 2000; Salem and Powers, 2001; Crossley et al., 2011; Yu et al., 2019). Squatting, which is symmetric by nature, can help coordinate bilateral muscle activities and strengthen weaker muscles on one side (e.g., among hemiplegia patients) or both sides. Compared to walking, squatting is often perceived to be safer for patients who are unable to perform these activities independently. In addition, the range of motion and the joint torques required for squatting are often greater than walking (Yu et al., 2019). With a reliable lower extremity rehabilitation exoskeleton, performing squatting exercises without external help (e.g., from a clinician) will be a confidence boost for patients to use the exoskeleton independently. However, in order for the exoskeletons to cooperate with the human without causing risks of harm, advanced balance controllers to robustly perform squatting motion that can deal with a broad range of environment conditions and external perturbations need to be developed.

Most existing lower extremity rehabilitation exoskeletons require the human operator to use canes for additional support or having a clinician or helper to provide balance assistance to avoid falling down. They often offer assistance via pre-planned trajectories of gait and provide limited control to perform diverse human motions. Some well known exoskeletons include the ReWalk (ReWalk Robotics), Ekso (Ekso bionics), Indego (Parker Hannifin), TWIICE (Vouga et al., 2017) and VariLeg (Schrade et al., 2018). When holding the crutches, the patient's interactions with the environment is also limited (Baud et al., 2019). One example is that the patient is unlikely to perform squatting with long canes or crutches. Very few exoskeletons are able to assist human motions such as walking without the need of crutches or helpers with the exception of a few well known ones: the Rex (Rex Bionics) (Bionics, 2020) and the Atalante (Wandercraft) (Wandercraft, 2020). These exoskeletons free the user's hands, but come at the cost of an very low walking speed and an increased overall weight (38 kg for the Rex, 60 kg for the Atalante) and are very expensive (Vouga et al., 2017). In this paper, we introduce a relatively light weight lower extremity exoskeleton that includes a sufficient number of degrees of freedom (DoF) with strong actuation. On each side, this skeleton system has a one DoF hip flexion/extension joint, a one DoF knee flexion/extension joint, and a 2-DoF ankle joint, which can perform ankle dorsi/plantar flexion as well as inversion/eversion that can swing the center of mass laterally in the frontal plane. Moreover, four force sensors are equipped on each foot for accurate measurement of ground reaction forces (GRFs) to estimate the center of pressure (CoP) so as to build automatic balance control without external crutches assistance.

Designing a robust balance control policy for a lower extremity exoskeleton is particularly important and represents a crucial challenge due to the balance requirement and safety concerns for the user (Chen et al., 2016; Kumar et al., 2020). First, the control policy needs to run in real-time with limited sensing and capabilities dictated by the exoskeletons. Second, due to various levels of patients' disability, the human-exoskeleton interaction forces are unpredictable and could vary substantially and cause conventional motion controllers to behave unreliably or the robot to fall down. Virtual testing of a controller with realistic human interactions in simulations is very challenging and the risk of testing on real humans is even greater. To the best of our knowledge, investigations presenting robust controllers against large and uncertain perturbation forces (e.g., due to human interactions) have rarely been carried out as biped balance control without perturbation itself is a challenging task. Most existing balance controller designs for such lower extremity rehabilitation exoskeletons focused mostly on the trajectory tracking method, conventional control like Proportional–Integral–Derivative (PID) (Xiong, 2014), model-based predictive control (Shi et al., 2019), fuzzy control (Ayas and Altas, 2017), impedance control (Hu et al., 2012; Karunakaran et al., 2020), and momentum-based control for standing (Bayon et al., 2020). Although the trajectory tracking approaches can be easily applied to regular motions, its robustness against unexpected large perturbations is not great. On the other hand, model-based predictive control could be ineffective or even unstable due to inaccurate dynamics modeling, and it typically requires a laborious task-specific parameters tuning. The momentum-based control strategies have also been applied to impose standing balancing on the exoskeleton (Bayon et al., 2020; Emmens et al., 2018), which was first applied in humanoid robotics to impose standing and walking balance (Lee and Goswami, 2012; Koolen et al., 2016). This method aimed to simultaneously regulate both the linear and angular component of the whole body momentum for balance maintenance with desired GRF and CoP at each support foot. The movement of system CoP is an important indicator of system balance (Lee and Goswami, 2012). When the CoP leaves or is about to leave the support base, a possible toppling of a foot or loss of balance is imminent and the control goal is to bring the CoP back inside the support base to keep balance and stability. Although the CoP information can sometimes be estimated from robot dynamics (Lee and Goswami, 2012), the reliability of such estimation highly depends on the accuracy of the robot model and sensing of joint states. When a human user is involved, it is almost impossible to estimate the CoP accurately due to the difficulty to estimate the user's dynamic properties or real-time joint motions. Therefore, it is highly desired to obtain the foot CoP information directly and accurately. In most existing lower extremity rehabilitation exoskeletons, the mechanical structures of the foot are either relatively simple with no force or pressure sensors or with sensors but no capability to process the GRF and CoP information for real-time fall detection or balance control. In this work, a lower extremity rehabilitation exoskeleton with force sensors equipped on each foot for accurate estimation of CoP is presented. Inspired by the CoP-associated balance and stability (Lee and Goswami, 2012), this paper aims to explore a robust motion controller to encourage the system CoP to stay inside a stable region when subjected to the uncertainty of human interaction and perturbations.

Recently, model-free control methods, like reinforcement learning (RL), promise to overcome the limitations of prior model-based approaches that require an accurate dynamic model. It has gained considerable attention in multi-legged robots control for their capability to produce controllers that can perform a wide range of complicated tasks (Peng et al., 2016; Peng and van de Panne, 2017; Peng et al., 2018; Hwangbo et al., 2019; Peng et al., 2020). The RL-based balance control approach for lower extremity rehabilitation exoskeletons to perform squatting motion have not been investigated before, especially when balancing with a human strapped inside is considered. Since the coupling between the human and exoskeleton could leads to unexpected perturbation forces, it is highly desired to develop a robust controller to learn collaborative human-robot squatting skills. In this paper, we propose a novel robust control framework based on RL to train a robust control policy that operates on the exoskeleton in real-time so as to overcome the external perturbations and unpredictable varying human-exoskeleton force.

The central contributions of this work are summarized in the following:

- We build a novel RL-based motion control framework for a lower extremity rehabilitation exoskeleton to imitate realistic human squatting motion under random adversary perturbations or large uncertain human-exoskeleton interaction forces.
- We take advantage of the foot CoP information by incorporating it into the state input of the control policy as well as the reward function to produce a balance controller that is robust against various perturbations.
- We demonstrate that the lightweight exoskeleton can carry a human to perform robust and well-balanced squatting motions in a virtual environment with an integrated exoskeleton and full-body human musculoskeletal model.

To demonstrate the effectiveness and robustness of the proposed control framework, a set of numerical experiments under external random perturbations and varying human-exoskeleton interactions are conducted. Dynamics randomization is incorporated into the training to minimize the effects of model inaccuracy and prepare for sim-to-real transfer.

2 EXOSKELETON AND INTERACTION MODELING

2.1 Mechanical Design of a Lower Extremity Robotic Exoskeleton

A lower extremity robotic exoskeleton device (Androwis et al., 2017) is currently under development by the authors to assist patients with ADL, such as balance, ambulation and gait rehabilitation. In **Figure 1A**, the physical prototype of this exoskeleton is shown. The total mass of the exoskeleton is $20.4kg$ and the frame of the exoskeleton has been manufactured with Onyx (Markforged's nylon with chopped fiber) reinforced by continuous carbon fiber between layers, using Markforged's Mark Two printer (Markforged, INC., MA). The exoskeleton has 14 independent DoFs (including six global DoFs for the pelvis root joint): one DoF for the hip flexion/extension joint, one DoF for the knee flexion/extension joint, and two DoFs for the ankle dorsiflexion/plantarflexion and inversion/eversion joint on each side of the body. The 6-DoF pelvis (root) joint is a free joint and unactuated, and the rest eight DoFs on both sides are actuated. Unlike most commercial rehabilitation exoskeletons that have either passive or fixed ankles, the ankle of our system includes powered 2-DoF to assist with dorsiflexion/plantarflexion and inversion/eversion (Nunez et al., 2017). All joints of the robotic exoskeleton are powered by smart actuators (Dynamixel Pro Motor H54-200-S500-R).

Both hip and knee joints are driven by bevel gears for compact design. The ankle is actuated by two parallel motors that are attached to the posterior side the shank support and operate simultaneously in a closed-loop to flex or abduct the ankle. **Figure 1C** shows the foot model of our exoskeleton system. At the bottom of each foot plate, four 2000N 3-axis force transducers (OptoForce Kft, Hungary) are installed to measure GRFs. These measured forces can be used to determine the CoP in real-time. The CoP is the point on the ground where the tipping moment acting on the foot equals zero (Sardain and Bessonnet, 2004), with the tipping moment defined as the moment component that is tangential to the foot support surface. Let O_i denote the location of the $i-th$ force sensor and F_i be its measured force, n denote the unit normal vector of the foot support surface, and C denote the CoP point where the tipping moment vanishes:

$$\left[\sum_{i}^{4}(O_iC\times F_i)\right]\times n=0 \tag{1}$$

from which C can be computed.

2.2 Exoskeleton Modeling

A multibody dynamics model of the exoskeleton, shown in **Figure 1A**, is created with mass and inertia properties estimated from each part's 3D geometry and density or measured mass. Simple revolute joints are used for the hips and knees. Each ankle has two independent rotational DoFs but their rotation axes are located at different positions (**Figure 1C**). These two DoFs are physically driven by the closed-loop of two ankle motors together with linkage of universal joints and screw joints. When extending or flexing the ankle, both motors work in sync to move the two long screws up or down simultaneously. When adducting or abducting the ankle, the motors move the screws in opposite directions. The motors can generate up to 160*Nm* dorsi/plantar flexion torque (Nunez et al., 2017). In this study, we do not directly control these ankle motors. Instead, we assume the two ankle DoFs can be independently controlled with sufficient torques from these motors.

2.3 Modeling of Human Exoskeleton Interactions

2.3.1 Human Musculoskeletal Model

To simulate realistic human exoskeleton interactions, the exoskeleton is integrated with a full body human musculoskeletal model (Lee et al., 2019) that has a total mass of 61*kg* and includes 284 musculotendon units. The integrated human and lower extremity exoskeleton model is shown in **Figure 1D**. The muscle geometry is designed as a polyline that starts at the origin of the muscle, passes through a sequence of way points, and ends at the insertion (Lee et al., 2019). The muscle dynamics is simulated using a Hill-type model (Thelen, 2003; Delp et al., 2007), which includes the muscle force-velocity and force-length relations as well as the muscle activation as follows:

$$F = \left[a \cdot F_L(l) \cdot F_V(\dot{l}) + F_P(l)\right] \times F_{max} \tag{2}$$

where F_{max} is the maximum isometric muscle force, $a \in [0, 1]$ is the muscle activation, l is the normalized muscle length. F_L and F_V are normalized force-length and force-velocity functions, respectively. The normalized passive force-length relationship of muscle F_P is represented by an exponential function:

$$F_P = \frac{e^{k^P\left(l^M-1\right)/\varepsilon_0^M} - 1}{e^{k^P} - 1} \tag{3}$$

where k^P is an exponential shape factor and set equal to four, and ε_0^M is a fixed passive muscle strain (at which point $F_P = 1$) and set to 0.6. When the muscle is fully passive (zero activation or without active muscle contraction), the muscle can only develop passive force $F_P \times F_{max}$ because of its background elasticity due to stretch. In this paper, we do not consider the active muscle contraction of the human operator, considering the operator could be a patient suffering from spinal cord injury or stroke, with very limited or no control of his or her own body. Nonetheless, the passive muscle forces $F_P \times F_{max}$ produced from all musculotendon units during movement are incorporated in the training to influence the human dynamics.

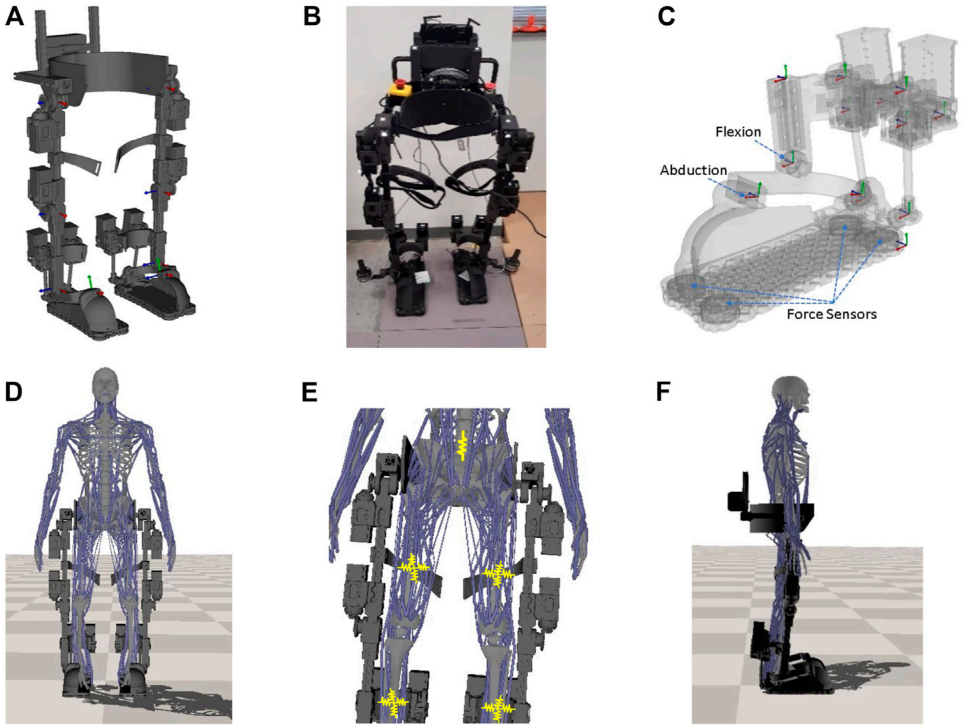

FIGURE 1 | The lower extremity exoskeleton and integrated human-exoskeleton model. **(A)** The multibody dynamics model of the exoskeleton with joint frames (red/green/blue axes) rendered in place. **(B)** The physical prototype of the exoskeleton. **(C)** The foot model with force sensors locations and the two independent ankle DoFs indicated. The joint axes of the model are displayed as well. **(D)** Integrated human and lower extremity exoskeleton model. **(E)** Spring (yellow) connections between the human and the strap on the robot. **(F)** Side view.

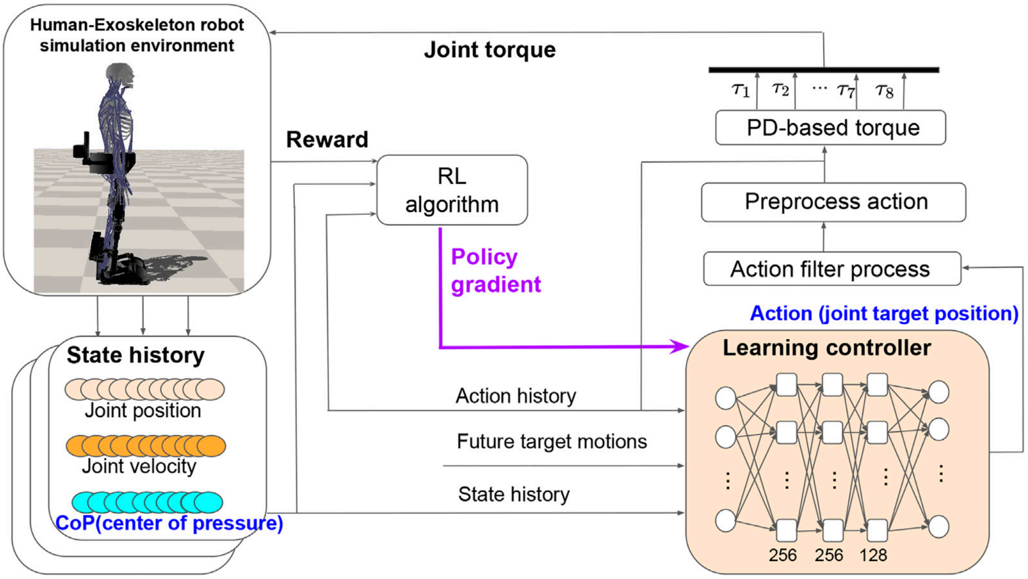

FIGURE 2 | The overall learning process of the integrated robust motion controller. We construct the learning controller as a Multi-Layer Perception (MLP) neural network that consists of three hidden layers. The neural network parameters are updated using the policy gradient method. The network produces joint target positions, which are translated into torque-level commands by PD (Proportional Derivative) control.

TABLE 1 | Hyper-parameters settings for training.

Parameters	Value	Parameters	Value
Discount factor	0.99	Epochs	10
Policy adam learning rate	10^{-4}	Clip threshold	0.2
Batch size	128	Memory buffer	2048

2.3.2 Human-Exoskeleton Interactions

To integrate the human musculoskeletal model with the exoskeleton, the root of the human model (pelvis) is attached to the exoskeleton hip through a prismatic joint that only allows relative movement along the vertical (up and down) direction (all rotations and the translations along the lateral and fore-and-aft directions are fixed). The assembled exoskeleton, as shown in **Figure 1D**, has straps around the hip, femur and tibia to constraint the human motion. Here we utilize the spring models to simulate the strap forces. **Figure 1E** shows the spring connections between the human and exoskeleton at the strap locations with the yellow zigzag lines illustrating the springs. From **Figure 1E**, an elastic spring with stiffness $k_h = 10000N/m$ is utilized to generate a vertical force and simulate the interaction between the human pelvis and the robot waist structure. At each femur strap location, we use four springs (arranged in 90° apart) with stiffness $k_f = 2000N/m$ to simulate the connections. Similar setup with stiffness $k_t = 2000N/m$ are used for the tibia strap connection. We choose these spring stiffness values based on empirical testing. These selected stiffness values produce a tight connection such that it can generate a reasonable (small) relative movement between the human and exoskeleton. During movement, the spring models can generate varying interaction forces between the human and exoskeleton robot during motion, which exert forces on both human and exoskeleton. We assume there is no relative motion between the human foot and the exoskeleton foot due to tight coupling and model that as a welding constraint. For simplicity, the joints at the upper limb of the human are fixed as weld joints.

3 LEARNING CONTROLLER FOR BALANCED MOTION

In this section, we propose a robust motion control training and testing framework based on RL that enables the exoskeleton to learn squatting skill with strong balance and robustness. **Figure 2** shows the overall learning process of the motion controller. The details of the motion controller learning process will be introduced in the following sections.

3.1 Reinforcement Learning With Motion Imitation

The controller (or control policy) is learned through a continuous RL process. We design the control policy through a neural network with parameters θ, denoting the weights and bias in the neural network. The control policy can be expressed as $\pi_\theta(a|s)$ and the agent (neural network model) learns to update the parameters θ to achieve the maximum reward. The learning controller (i.e., control policy network) is implemented as a Multi-Layer Perception (MLP) network that consists of three fully connected layers and ReLU as the activation function, as illustrated in **Figure 2**, the sizes of three layers are set to 256, 256 and 128, respectively. At every time step t, the agent observes the state s_t from the environment, and selects an action a_t according to its control policy $\pi(a_t|s_t)$. The control policy network $\pi(a|s)$ is in the form of the probability distribution of actions in a given state. This action distribution is modeled as a Gaussian, with a state dependent mean $\mu(s)$ defined by the network, and a standard deviation (STD) Σ that is learned as the neural network parameters:

$$\pi(a|s) = \mathcal{N}(\mu(s), \Sigma) \tag{4}$$

The action a_t is sampled from this distribution, and then converted to control commands that drive the exoskeleton in the environment, which results a new state s_{t+1} and a scalar reward r_t immediately. The objective is to learn a control policy that maximizes the discounted sum of reward:

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}_{\tau\sim \mathrm{p}(\tau|\pi)}\left[\sum_{\mathrm{t}=0}^{\mathrm{T}-1} \gamma^{\mathrm{t}} \mathrm{r}_{\mathrm{t}}\right] \tag{5}$$

where $\gamma \in (0,1)$ is the discount factor, τ is the trajectory $\{(s_0, a_0, r_0), (s_1, a_1, r_1), \cdots\}$ and $p(\tau|\pi)$ denotes the likelihood of a trajectory τ under a given control policy π, and T is the horizon of an episode.

The control policy is task- or motion-specific and starts with motion imitation. Within the learning process (**Figure 2**), the input of the control policy network is defined by $s = \{p_{t-2:t}, v_{t-2:t}, c_{t-2:t}, a_{t-3:t-1}, \widehat{p}_{t+1:t+6}\}$, in which p and v are joint positions and velocities of the exoskeleton, and $a_{t-3:t-1}$ represents the action history of three previous steps. To learn a particular skill, we utilize the corresponding target joint pose from the task motion at six future time-steps $\widehat{p}_{t+1:t+6}$ as the motion prior for feasible control strategies. Considering the importance of CoP as an indicator of system balance and its ready availability from our exoskeleton design, we incorporate the CoP position history $c_{t-2:t}$ into the state as a feedback to the controller. As summarized in **Figure 2**, the learning controller is given as input the combination of the state history, the action history and the future target motions and outputs the joint target positions as the actions. The use of task motion data alleviates the need to design task-specific reward functions and thereby facilitates a general framework to learn a diverse array of behaviors.

3.2 Learning With Proximal Policy Optimization

To train the control policy network, we use a model-free RL algorithm known as Proximal Policy Optimization (PPO). An effective solution to many RL problems is the family of policy gradient algorithms, in which the gradient of the expected return with respect to the policy parameters is computed and used to update the parameters θ through gradient ascent. PPO is a model-free policy gradient algorithm that samples data through interaction with the environment and optimizes a "surrogate" objective function (Schulman et al., 2017). It utilizes a trust region constraint to force the control policy update to ensure that the new policy is not too far away from the old policy. The probability ratio $r_t(\theta)$ is defined by:

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}. \tag{6}$$

This probability ratio is a measure of how different the current policy is from the old policy $\pi_\theta old$ (the policy before the last update). A large value of this ratio means that there is a large change in the updated policy compared to the old one. PPO also introduces a modified objective function that adopts clipped probability ratios which forms a pessimistic estimate of the policy's performance and avoids a reduction in performance during the training process. The "surrogate" objective function is described by considering the clipped objective:

$$L(\theta) = \mathbb{E}_t\left[min\left(r_t(\theta)\widehat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\widehat{A}_t\right)\right] \tag{7}$$

where ϵ is a small positive constant which constrain the probability ratio $r_t(\theta)$. $\widehat{A}_t$ denotes the advantage value at time step t. $clip(\cdot)$ is the clipping function. Clipping the probability ratio discourages the policy from changing too much and taking the minimum results in using the lower, pessimistic bound of the unclipped objective. Thus any change in the probability ratio is included when it makes the objective worse, and otherwise is ignored. This can prevent the policy from changing too quickly and leads to more stable learning. The control policy can be updated by maximizing the clipped discounted total reward in **Eq. 7** with a gradient ascent.

3.3 Proportional-Derivative-Based Torque

It has been reported that the performance of RL for continuous control depends on the choice of action types (Peng and van de Panne, 2017). It works better when the control policy outputs PD position targets rather than joint torque directly. In this study, we also let the learning controller output the joint target positions as

TABLE 2 | Dynamic parameters and their respective range of values used during training and testing. A larger range of values are used during testing to evaluate the generalization ability of control policies in dynamics uncertainties.

Dynamic parameters	Training range	Testing range
Friction coefficient	[0.9,1.6]*default value	[0.7,2.0]*default value
Mass	[0.8,1.2]*default value	[0.7,1.5]*default value
Motor strength	[0.8,1.2]*default value	[0.7,1.3]*default value
Observation latency	[0,0.04]s	[0,0.06]s
Inertial	[0.5,1.5]*default value	[0.4,1.6]*default value
Center of Mass	[0.9,1.2]*default value	[0.8,1.3]*default value

actions (shown in **Figure 2**). To obtain smooth motions, actions from the control policy network are first processed by a second low-pass filter before being applied to the robot. Our learning process allows the control policy network and the environment to operate at different frequencies since the environment often requires a small time step for integration. During each time integration step, we apply preprocessed actions that are linear interpolated from two consecutive filtered actions. Then the preprocessed actions are specified as PD targets and the final PD-based torques applied to each joint are calculated as

$$\tau = clip\left(k_p\left(a_t - p_t\right) - k_v\dot{p}_t, -\widehat{\tau}, \widehat{\tau}\right) \tag{8}$$

where k_p and k_v are proportional gain and differential gain, respectively. The function $clip(\cdot)$ returns the upper bound $\widehat{\tau}$ or the lower bound$-\widehat{\tau}$ if the torque τ exceeds the limit.

3.4 Reward Function

We design the reward function to encourage the control policy to imitate a target joint motion $\widehat{p}_t$ of the exoskeleton while maintaining balance with robustness. The reward function consists of pose reward r_t^p, velocity reward r_t^v, end-effector reward r_t^e, root reward r_t^{root}, center of mass reward r_t^{com}, CoP reward r_t^{cop}, and torque reward r_t^{torque}, which is defined by:

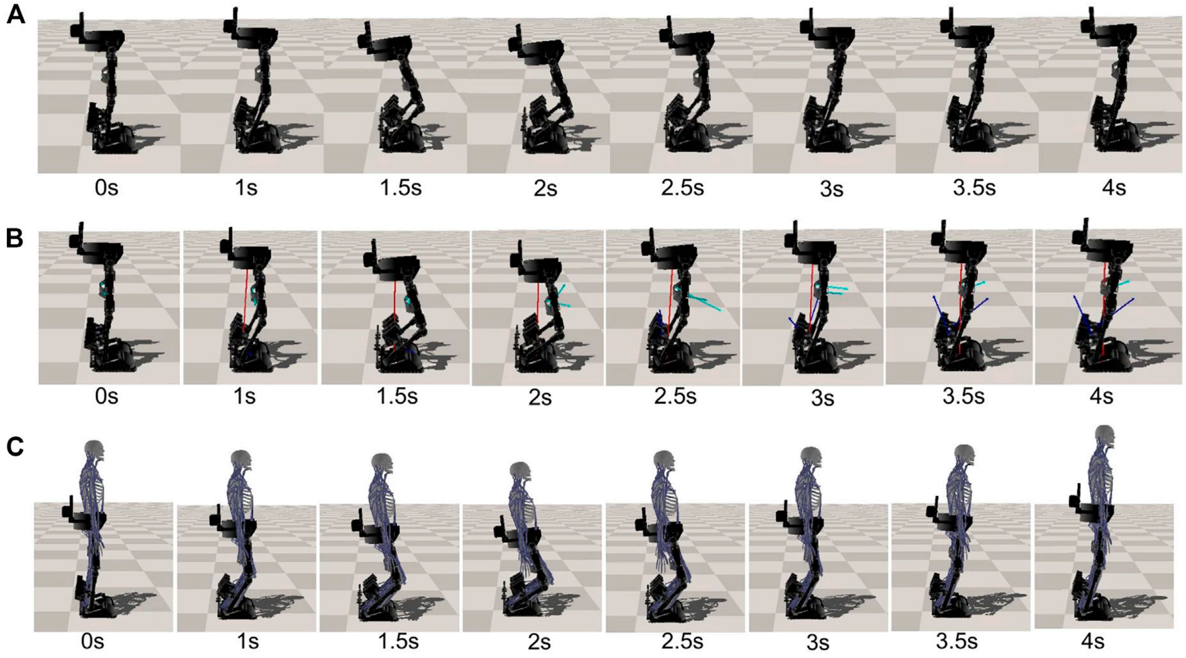

FIGURE 3 | Snapshots of the squatting motion of the exoskeleton. This RL-based controller enables the lower extremity exoskeleton to perform the squatting skill under different types of external perturbations with various intensities. **(A)** Performing the squatting skill without perturbation forces. **(B)** Performing the squatting skill with large random perturbation forces. (Red, cyan and blue arrows show the random perturbation force applied on the hip, femur and tibia, respectively). **(C)** Performing the squatting skill with human interaction.

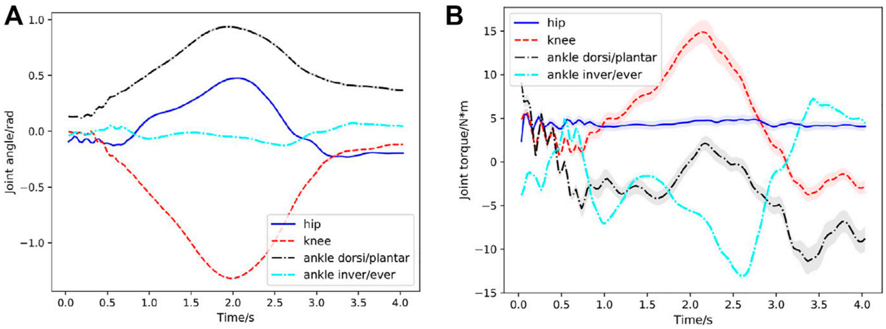

FIGURE 4 | Case1: Joint behavior statistics in the first squatting cycle (curve: mean; shade: STD) with respect to time for 200 simulated environments. **(A)** Hip, knee, ankle dorsiflexion/plantarflexion, ankle inversion/eversion joint angles with respect to time during the first squatting cycle. Since the learned STDs Σ (in **Eq. 4**) of the joint angles are very small (less than 0.05), the variances of joint angles are indiscernible in figure. **(B)** Corresponding joint torques during the first squatting cycle.

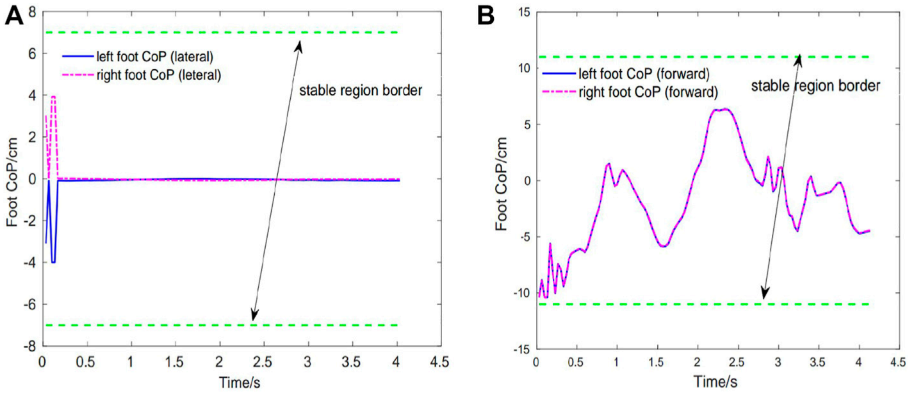

FIGURE 5 | Case1: Foot CoP trajectories during the first squatting cycle. Green dotted line depicts the stable region border of CoP. **(A)** Left foot CoP and right foot CoP trajectories (lateral direction). **(B)** Left foot CoP and right foot CoP trajectories (forward direction).

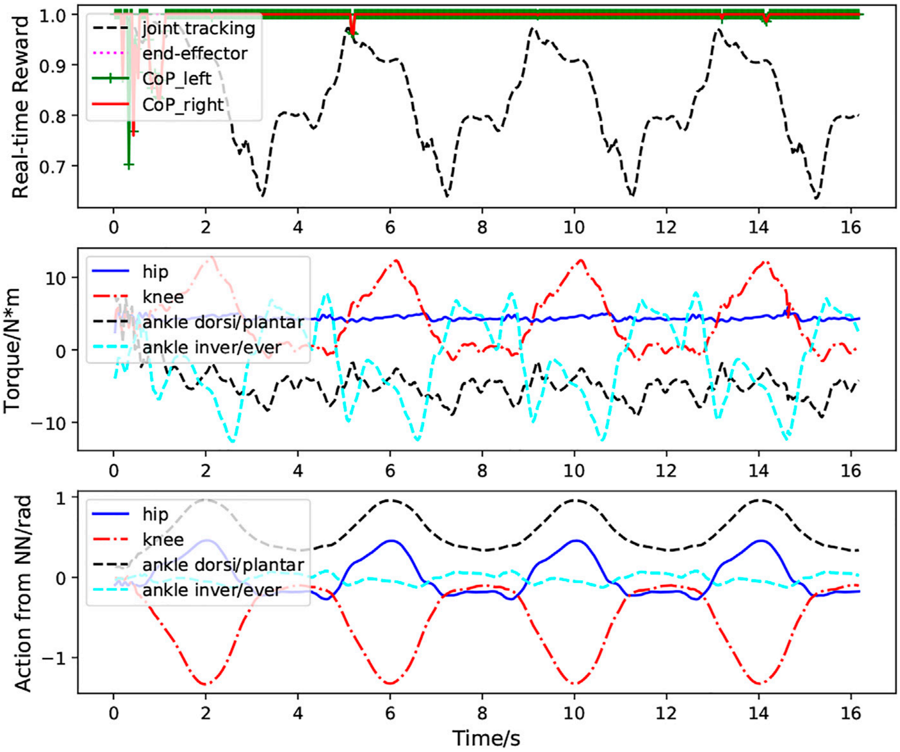

FIGURE 6 | Case1: Performance of the RL based controller without external perturbation during multiple squatting cycles. The first figure demonstrates the foot CoP reward (green and red lines), end-effector tracking reward (magenta line) and joint position tracking reward (black line) calculated according to **Eqs 10–15**. The bottom figure depicts the actions for the hip, knee ankle joint predicted from the neural network.

$$r_t = w^p r_t^p + w^v r_t^v + w^e r_t^e + w^{root} r_t^{root} + w^{com} r_t^{com} + w^{cop} r_t^{cop} + w^{torque} r_t^{torque} \tag{9}$$

where $w^p, w^v, w^e, w^{root}, w^{com}, w^{cop}, w^{torque}$ are their respective weights. The pose reward r_t^p and velocity reward r_t^v match the current and task (target) motions in terms of the joint positions p_t and velocities $\dot{p}_t$:

$$r_t^p = \exp\left[-\sigma_p \sum_j \left\|\hat{p}_t^j - p_t^j\right\|^2\right] \tag{10}$$

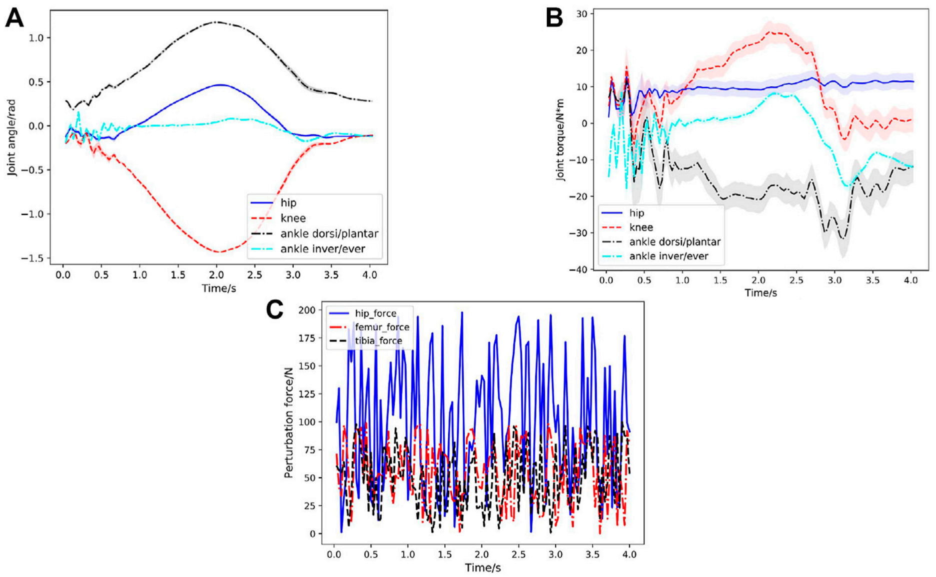

FIGURE 7 | Case2: Joint behavior statistics (curve: mean; shade: STD) under random perturbation forces in the first squatting cycle with respect to time for 200 simulated environments. **(A)** Hip, knee, ankle dorsiflexion/plantarflexion, ankle inversion/eversion joint angles with respect to time in the first squatting cycle. The learned STDs Σ (in **Eq. 4**) of the joint angles are very small (less than 0.08). **(B)** Corresponding joint torques in the first squatting cycle. **(C)** Magnitudes of the random perturbation forces applied on the three parts of the exoskeleton: hip, femur and tibia during testing.

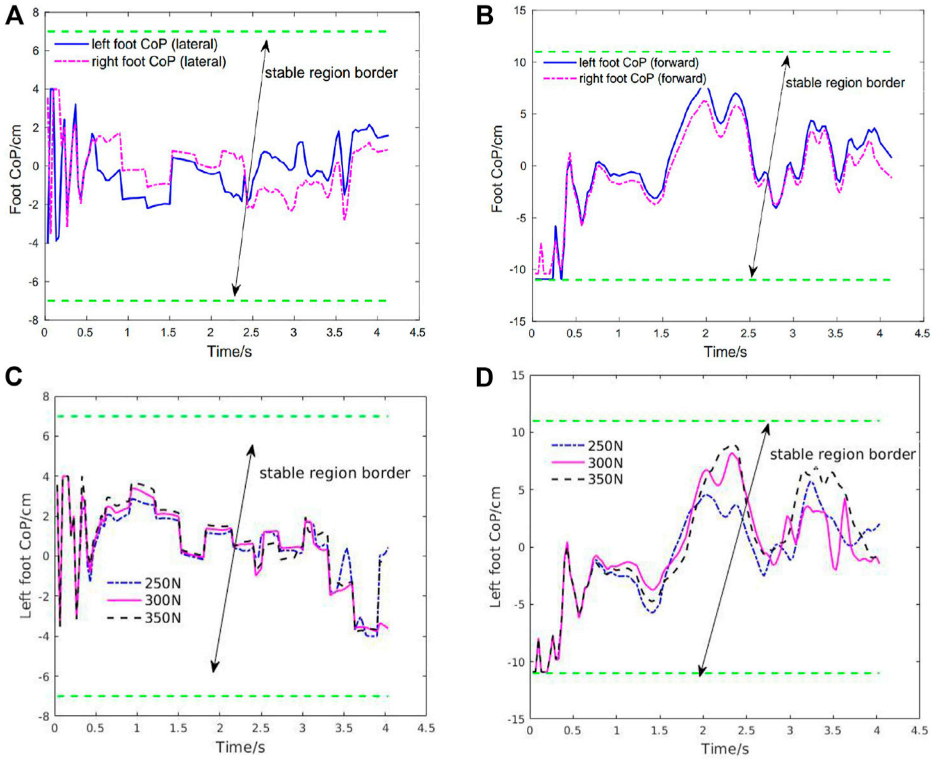

FIGURE 8 | Case2: Foot CoP trajectories under random external perturbations in the first squatting cycle and robustness test on the foot CoP stability under 75% greater perturbation forces compared with the training setting. **(A)** Left and right foot CoP trajectories (lateral direction). **(B)** Left and right foot CoP trajectories (forward direction). **(C)** Left foot CoP trajectory (lateral direction) under 75% greater perturbation forces compared with the training setting. **(D)** Left foot CoP trajectory (forward direction) under greater perturbation forces compared with the training setting.

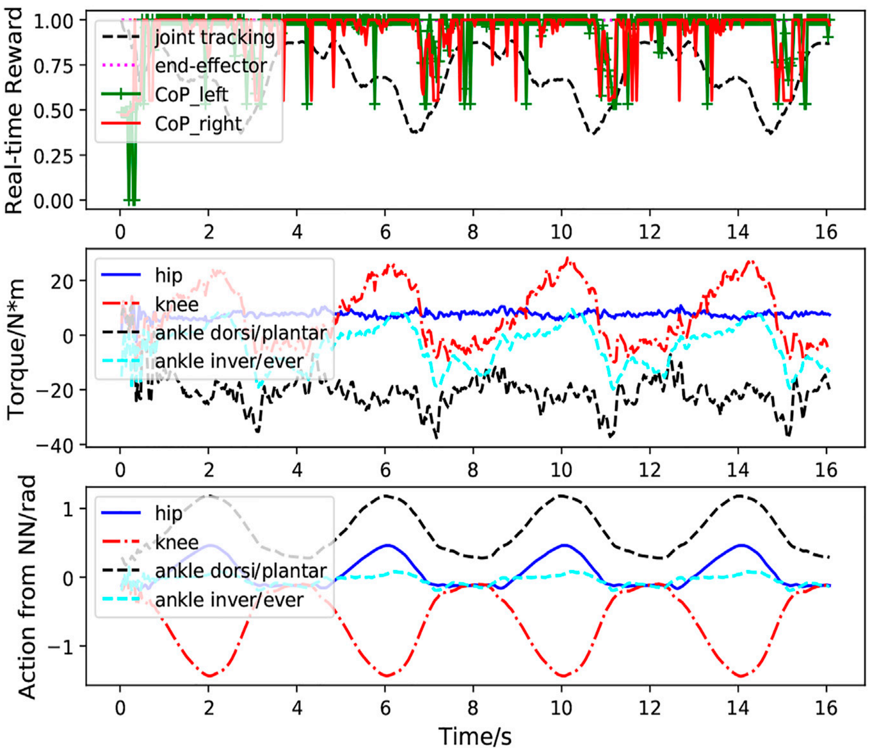

FIGURE 9 | Case2: Real-time performance of the reinforcement-learning based controller with large, random external perturbation during multiple squatting cycles. The first figure demonstrates the real-time foot CoP reward (green and red lines), end-effector tracking reward (magenta line) and joint position tracking reward (black line) calculated according to **Eqs 10–15**. The bottom figure depicts the actions for the hip, knee ankle joint predicted from the neural network.

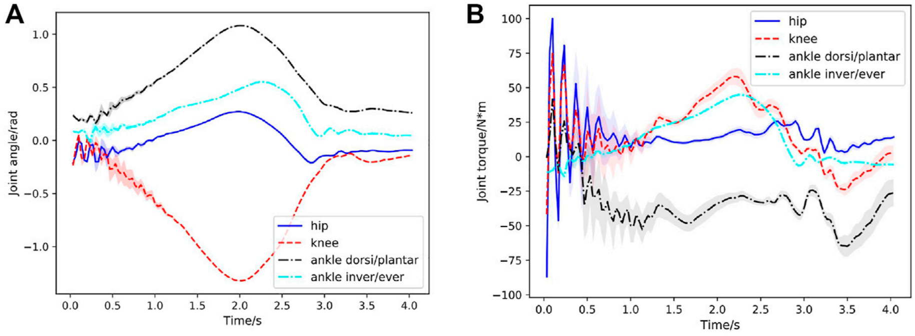

FIGURE 10 | Case3: Joint behavior statistics under human-exoskeleton interactions during the first squatting cycle with respect to time for 200 simulated environments. **(A)** Hip, knee, ankle dorsiflexion/plantarflexion, ankle inversion/eversion joint angles in the first squatting cycle. The learned STDs Σ (in **Eq. 4**) of the joint angles are very small (less than 0.06). **(B)** Corresponding joint torques in the first squatting cycle.

$$r_t^v = \exp\left[-\sigma_v \sum_j \left\|\widehat{\dot{p}}_t^j - \dot{p}_t^j\right\|^2\right] \tag{11}$$

where j is the index of joints, $\widehat{p}_t^j$ and $\widehat{\dot{p}}_t^j$ are target position and velocity, respectively. The end-effector reward is defined to encourage the robot to track the target positions of selected end-effectors:

$$r_t^e = \exp\left[-\sigma_e \sum_i \left\|\widehat{x}_t^i - x_t^i\right\|^2\right] \tag{12}$$

where i is the index of the end-effector. Let x_t^i be the position of end-effectors, that include left foot and right foot, relative to the moving coordinate frame attached to the root (waist structure). The end-effectors motions are supposed to match well if the joint angles match well, and vice versa.

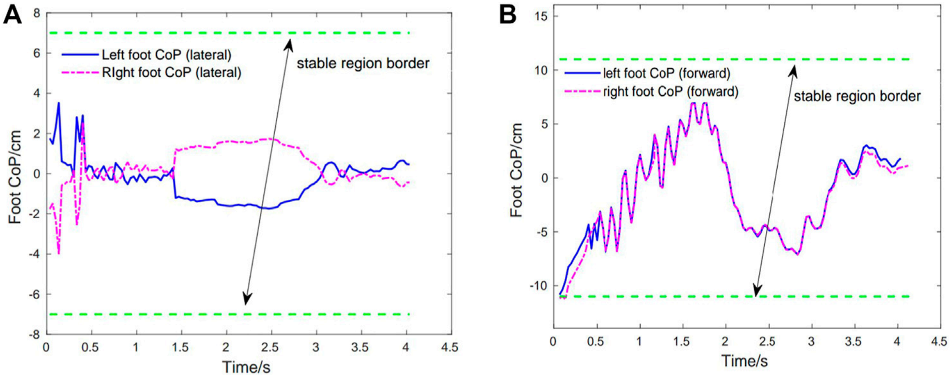

FIGURE 11 | Case3: Foot CoP trajectories under human-exoskeleton interactions in the first squatting cycle. The stable region border is marked with green dotted lines. **(A)** Left foot CoP and right foot CoP trajectories (lateral direction). **(B)** Left foot CoP and right foot CoP trajectories (forward direction).

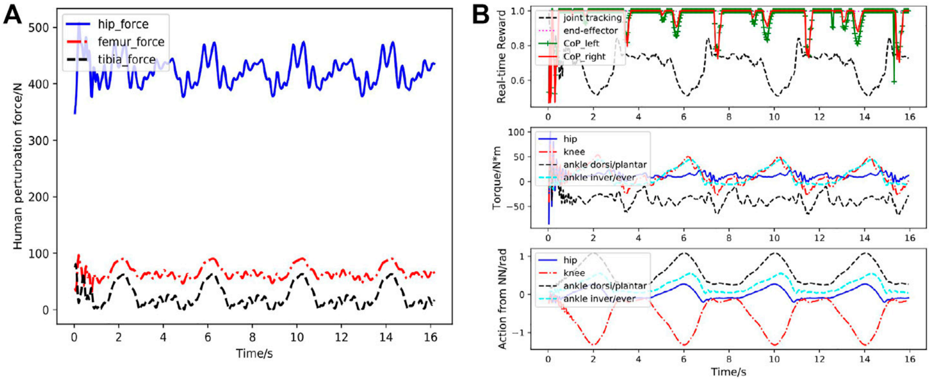

FIGURE 12 | Case3: Performance of the RL based controller under human-exoskeleton interactions during more squatting cycles. **(A)** Human-exoskeleton interaction (strap) forces during multiple squatting cycles. hip–force, femur–force and tibia–force in the figure represent the interaction forces between the human and the strap on the exoskeleton, respectively. **(B)** Performance of the RL based controller. The first figure demonstrates the foot CoP reward (green and red lines), end-effector reward (magenta line) and joint position tracking reward (black line) calculated according to **Eqs 10–15**. The bottom figure depicts the actions for the hip, knee ankle joint predicted from the neural network.

We also design the root reward function r_t^{root} to track the task root motion including the root's position $\widehat{x}_t^{root}$ and rotation $\widehat{q}_t^{root}$.

$$r_t^{root} = \exp\left[-\sigma_{r1}\left\|\widehat{x}_t^{root} - x_t^{root}\right\|^2 - \sigma_{r2}\left\|\widehat{q}_t^{root} - q_t^{root}\right\|^2\right] \tag{13}$$

The overall center of mass reward r_t^{com} is also considered in the learning process, enabling the control policy to track the target height during the complete squatting cycle.

$$r_t^{com} = \exp\left[-\sigma_{com}\left\|\widehat{x}_t^{com} - x_t^{com}\right\|^2\right] \tag{14}$$

The movement of system CoP is an important indicator of system balance as set forth in the introduction. When the CoP leaves or is about to leave the foot support polygon, a possible toppling of the foot or loss of balance is imminent. During squatting (or walking), there are always one or two feet on the ground and the support polygon on the touching foot persists, and thus the CoP criterion is highly relevant for characterizing the tipping equilibrium of a bipedal robot (Sardain and Bessonnet, 2004). When the CoP point lies within the foot support polygon, it can be assumed that the biped robot can keep balance during squatting. Considering the importance of CoP, we incorporate the CoP positions in the state input as a feedback from the controller and also add a CoP reward function to encourage the current CoP (c_t^{cop}) to stay inside a stable region around the center of the foot support. By considering the geometric of the foot in the lower extremity exoskeleton design, the stable region for foot CoP is defined as a rectangle area S around the foot geometric center whose width and length are set to $7cm$ and $11cm$, respectively (narrower in the lateral direction than forward direction). And the CoP reward function is expressed as

$$r_t^{cop} = \begin{cases} \exp\left[-\sigma_{cop}\left\|D\left(c_t^{cop}, S\right)\right\|^2\right], \text{if } c_t^{cop} \in S \\ 0, \text{if } c_t^{cop} \notin S \end{cases} \tag{15}$$

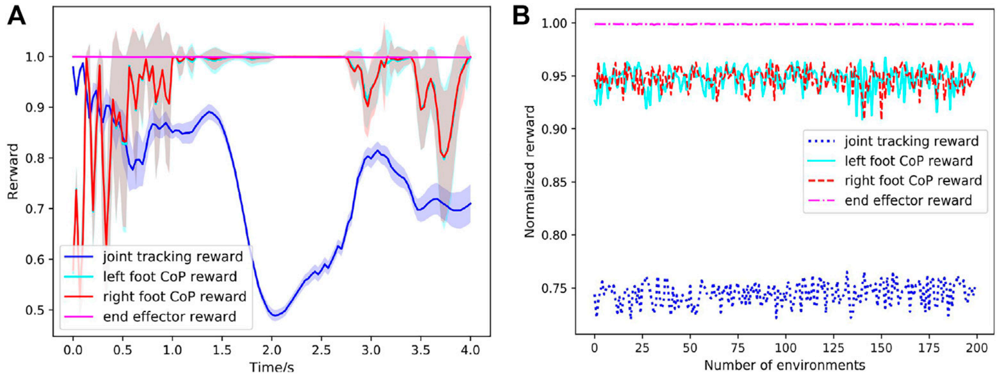

FIGURE 13 | Performance of learned controller under human-exoskeleton interactions in 200 simulated environments with different dynamics. The dynamic model of the exoskeleton is randomly initialized. **(A)** Rewards statistics (curve: mean; shade: STD) with respect to time for 200 simulated environments. **(B)** Average reward of a complete squatting cycle for each simulated environment.

where $D(\cdot,\cdot)$ is the Euclidean distance between CoP and the center of S. The goal of this CoP reward is to encourage the controller to predict an action that will improve the balance and robustness of the exoskeleton's motion.

At last, we design the torque reward to reduce energy consumption or improve efficiency and to prevent damaging joint actuators during the deployment.

$$r_t^{torque} = \exp\left[-\sigma_{torque}\sum_i \|\tau_i\|^2\right] \tag{16}$$

where i is the index of joints.

3.5 Dynamics Randomization

Due to the model discrepancy between the physics simulation and the real-world environment, well-known as reality or sim-to-real gap (Yu et al., 2018), the trained control policy usually performs poorly in the real environment. In order to improve the robustness of the controller against model inaccuracy and bridge the sim-to-real gap, we need to develop a robust control policy capable of handling various environments with different dynamics characteristics. To this end, we adopt dynamics randomization (Sadeghi and Levine, 2016; Tobin et al., 2017) in our training strategy, in which dynamics parameters of the simulation environment are randomly sampled from an uniform distribution for each episode. The objective in **Eq. 5** is then modified to maximize the expected reward across a distribution of dynamics characteristics $\rho(\mu)$:

$$\pi^\star = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}_{\mu\sim\rho(\mu)}\mathbb{E}_{\tau\sim \mathrm{p}(\tau|\pi,\mu)}\left[\sum_{\mathrm{t}=0}^{\mathrm{T}-1}\gamma^{\mathrm{t}}\mathrm{r}_{\mathrm{t}}\right], \tag{17}$$

where μ represents the values of the dynamics parameters that are randomized during training. By training policies to adapt to variability in environment dynamics, the resulting policy will be more robust when transferred to the real world.

4 NUMERICAL EXPERIMENT RESULTS AND DISCUSSION

We design a set of numerical experiments aiming to answer the following questions: 1) Can the learning process generate feasible control policies to control the exoskeleton to perform well-balanced squatting motions? 2) Will the learned control policies be robust enough under large random external perturbation? 3) Will the learned control policies be robust enough to sustain stable motions when subjected to uncertain human-exoskeleton interaction forces from a disabled human operator?

4.1 Simulation and Reinforcement Learning Settings

To demonstrate the effectiveness of our RL-based robust controller, we train the lower extremity exoskeleton to imitate a 4s reference squatting motion that is manually created based on human squatting motion. The reference squatting motion can provide guidance for motion mimicking but needs not to be generated precisely. The exoskeleton contains eight joint DoFs actuated with motors, all of which are controlled by the RL controller. On each leg, there are has four actuators, including one actuator for hip flexion/extension, one actuator for knee flexion/extension, two actuators for ankle dorsiflexion/plantarflexion and ankle inversion/eversion, respectively. The open source library DART (Lee et al., 2018) is utilized to simulate the exoskeleton dynamics. The GRFs are computed by a Dantzig LCP (linear complementary problem) solver (Baraff, 1994). We utilize PyTorch (Paszke et al., 2019) to implement the neural network and the PPO method for the learning process. The networks are initialized by the Xavier uniform method (Glorot and Bengio, 2010). We use a desktop computer with an Intel® Xeon(R) CPU E5-1660 v3 at 3.00 GHz × 16 to generate samples in parallel during training. Totally about 20 million samples are collected in each simulation. The policy and value networks are updated at a learning rate of 10^{-4}, which is linearly decreased to 0 when 20 million samples are collected. PPO algorithm is robust in that hyperparameter initialization is a bit more forgiving and it can handle a wide variety of RL tasks. We do not deliberately tune hyperparameters and just use the common setting as in the literature (Schulman et al., 2017; Tan et al., 2018). Hyper-parameters settings for training using PPO are shown in **Table 1**.

To verify the robustness of the trained controller, we test the control policies in out-of-distribution simulated environments, where the dynamic parameters of the exoskeleton are sampled randomly from a larger range of values than those during training. **Table 2** shows the dynamics parameters details of the exoskeleton and their range during training and testing. Note that the observation latency denotes the observation time delay in the real physical system due to sensor noise and time delay during information transfer. Considering the observation latency improves the reality of the simulations and further increases the difficulty of policy training. The simulation frequency (time step for the environment simulation) and control policy output frequency are set to 900 and 30Hz, respectively. According to the PD torque **Eq. 8**, the parameters about the proportional gain k_p and differential gain k_v are set to 900 and 40, respectively. The differential gain k_v is chosen to be sufficiently high to prevent unwanted oscillation on the exoskeleton robot. From our experience, the control performance is robust against the variance of gains to a certain extent. For instance, increasing or decreasing the position gain k_p to 1200 or 800 does not noticeably change the control performance. We have tried different sets of rewards weights (in **Eq. 9**) in a proper range, we found the control performance with different sets of weights are similar, and we choose the best one: $w^p = 0.8, w^{cop} = 0.8, w^v = 0.1, w^{ee} = 0.7, w^{com} = 0.4, w^{root} = 0.7, w^{torque} = 0.1$. The torque limit for each joint is set to $100Nm$ in the simulation.

4.2 Learned Squatting Skill

4.2.1 Case 1–Feasibility Demonstration

In the first case, a squatting motion controller is learned from the 4s reference squatting motion without considering external perturbation. It is worth noting that, compared with the training, we use the larger-range dynamics randomization of the exoskeleton model to demonstrate the generalization ability of our learned controller (as shown in **Table 2**). A series of snapshots of the squatting behavior of the lower extremity exoskeleton under the learned control policy are shown in **Figure 3A**. The lower extremity exoskeleton can perform the squatting and stand-up cycle with a nearly symmetric motion. We test the learned controller in 200 out-of-distribution simulated environments, where the dynamics parameters are sampled from a larger range of values than those used during training (as shown in **Table 2**). **Figure 4A** displays the statistical results of the hip flexion/extension, knee flexion/extension, ankle dorsiflexion/plantarflexion and ankle inversion/eversion joint angles in the first squatting cycle. Joint torques statistics are depicted in **Figure 4B** and the peak torque at the knee joint for the squatting is around $13.5Nm$. From **Figure 4**, it is observed that the joint angles produced have very consistent mean values with near vanishing variances (as a result of very small STDs of the action distributions (**Eq. 4**) learned from the neural network). The joint torques also have good consistence in predicted mean torques with slightly higher variances, especially for the knee and ankle dorsi/plantar flexion joints. Due to the weights of the waist support structure and the battery (around $2.2kg$) mounted on the back of it, the hip joint consistently produces a positive torque to prevent the waist structure from rotating downwards due to gravity. The controller is able to achieve low average joint angle tracking error (about 1.22°) compared with the target squatting motion.

Figure 5 show the best obtained result of foot CoP trajectories (left and right feet) in the lateral and forward directions, which are calculated in real-time using the ground contact force information. As it can be seen, the exoskeleton controller can keep the foot CoP well inside the stable region for both lateral and forward directions in a complete squatting cycle. Noted at the beginning the CoP is close to the back edge (due to its initial state) but gradually it is bought to near the center. This indicates that the exoskeleton controller is able to recognize the current state of CoP and capable of bring it to a more stable state. And the balance in the lateral direction is better than that in the forward direction due to the symmetric nature of the squatting motion. The right foot CoP trajectories have very similar patterns with those of the left foot CoP.

Figure 6 presents the performance of the learned controller while performing multiple squatting cycles. We can clearly observe the high CoP rewards, indicating good system balance when the exoskeleton performs the squatting motion. The relatively high joint position tracking and end-effector tracking rewards illustrate strong tracking performance of the control system. The second figure shows the torques for the hip, knee, and ankle joints. The last figure demonstrates the predicted actions (PD target positions) for these joints. It is clear from these plots that the actions predicted from the policy network are smooth and exhibit clear cyclic patterns.

4.2.2 Case 2–Robustness Against Random Perturbation Forces

In the second case, we aim to verify the robustness of the controller under random external perturbation forces. From our tests, the learned control policy from case 1, trained without any perturbation forces, could perform well with random perturbation forces up to $100N$. To further improve the robustness of our controller to handle greater perturbation forces, we now introduce random perturbation forces during the training process of the learning controller. The perturbation forces are applied to three parts of the exoskeleton: hip, femur, and tibia. For femur and tibia, the magnitude of forces are randomly sampled in the range $(0, 100)N$ and no restriction is set for the direction. For the hip, we randomly sample the magnitude of force in $(0, 200)N$ but restricted the direction to $0 \sim 20$ degrees from the vertical direction assuming no large lateral pushing perturbation.

Figure 3B shows a series of snapshots of the lower extremity exoskeleton behavior with the newly trained controller when tested with random, large perturbation forces during squatting motion. Statistical results of the Joint angles and torques at the hip, knee and ankle joint in the first squatting cycle under 200 simulated environments are shown in **Figure 7**. We can clearly observe that the motion is still relative smooth and the torques calculated from **Eq. 8** have more ripples in response to the random perturbation forces. **Figure 7C** shows randomly varying perturbation forces applied on the hip, femur and tibia. Compared to the joint angles and torques without external perturbation (Case 1, **Figure 4**), the joint torques under random perturbation forces are almost doubled but the joint angles are relatively close. Under the large, random perturbation forces, the controller can achieve the target squatting motion with a low average joint angle tracking error (about 2.64°). Moreover, under the random perturbations that could influence the balance in the frontal plane, the learned controller can still keep balance and stability with the actuation of the ankle inversion/eversion joint.

Figure 8 shows the CoP trajectories of the left and right feet in both lateral and forward directions in the first squatting cycle. As shown in this figure, the foot CoP trajectories have oscillation under the random perturbation forces compared with Case1. But the robot can still keep the CoP inside the safe region of the support foot to guarantee stability and balance. It further validates that this controller enables the robot to perform squatting motion with strong stability and robustness. To further demonstrate the robustness of the learning controller, we increase the random perturbation forces up to 75% greater than the training perturbation (e.g., up to 350*N* for the hip force) and found that the exoskeleton can still perform the squatting motion without losing its balance as illustrated in **Figures 8C, D** about the left foot CoP stability. Real-time tracking results of the exoskeleton with multiple squatting cycles are shown in **Figure 9**. From **Figure 9**, both the foot CoP reward and the end-effector reward remain high under the random perturbation forces. The torques are greater than that without external perturbation forces, the peak torque for the knee joint is close to 40*Nm*. The actions predicted from the policy network are also smooth and exhibit cyclic patterns.

4.2.3 Case3–Robustness Under Human-Exoskeleton Interaction

In the third case, the human musculoskeletal model is integrated with the exoskeleton to simulate more realistic perturbation forces. As described in section 2.3, the springs at the strap locations can generate the varying interaction forces between the human and exoskeleton robot during the motion, which are applied on both human and exoskeleton. We first train the network with the integrated human-exoskeleton model to account for the interaction forces. Here we do not consider the active muscle contraction of the human operator or actuation torques on the human joints, considering the operator could be a patient suffering from spinal cord injury or stroke, with very limited or no control of his or her own body. Nonetheless, the passive muscle forces as described in **Eq. 3** during movement are incorporated. The squatting skill learned by the exoskeleton and performance of the motion controller are shown in **Figure 3C** and **Figures 10–13**.

As shown in **Figure 3C**, the rehabilitation exoskeleton is able to assist the human to perform the squatting motion without external assistance. Statistical results of angles and torques at the hip, knee and ankle joint of the left leg in the first cycle under 200 simulated environments are shown in **Figures 10A, B**. The torques at the hip, knee and ankle joint are greater than those without the human interaction while still below the maximum torques. The controller can still maintain low joint angle average tracking error (about 2.49°) under the varying human-exoskeleton forces. **Figure 11** shows the CoP trajectories of the left foot and right foot in the lateral and forward directions under the predicted human-skeleton interaction forces (as shown in **Figure 12A**). From **Figure 11A**, there exists a symmetric pattern between the left and right foot CoP trajectories in the lateral direction, which indicates that the squatting motion is well balanced in the frontal plane. Real-time tracking results of the controlled exoskeleton under the human-exoskeleton interactions with more squatting cycles are shown in **Figure 12**. As shown in **Figure 12B**, the high CoP reward in the first figure indicates that the proposed control system has strong stability and robustness to the human interaction forces. The peak torque for all joints are less than 70*Nm*. The largest peak torque happens at the ankle dorsi/plantar flexion joint (68*Nm*), which is much smaller than its 160*Nm* capacity (Nunez et al., 2017). The actions predicted from the policy network, shown in the last figure of **Figure 12**, are still smooth and cyclic in general.

Figure 13 visualizes the performance of the learned controller in 200 simulated environments with different dynamics. **Figure 13A** depicts the rewards statistics (mean and STD) with respect to time calculated from **Eqs 10–15** under 200 simulated environments. The end-effector reward indicating the foot tracking performance consistently maintains a high value, revealing the exoskeleton robot has no falling condition in 200 simulated environments with unfamiliar dynamics and it can stand on the ground with stationary feet when performing the complete squatting motion. The joint position tracking and foot CoP also achieve a high reward with less variance under more diverse dynamics of the exoskeleton robot. **Figure 13B** shows the average reward of a complete squatting cycle for each simulated environment. These results suggest that the learned controller is able to effortlessly generalize to environments that differ from those encountered during training and achieve good control performance under very diverse dynamics. The extensive testing performed with the integrated human-exoskeleton demonstrate that the RL controller is robust enough to sustain stable motions when subjected to uncertain human-exoskeleton interaction forces.

5 DISCUSSION

Through these designed numerical tests, we verified that our learning framework can produce effective neural network-based control policies for the lower extremity exoskeleton to perform well-balanced squatting motions. By incorporate adverse perturbations in training, the learned control policies are robust against large random external perturbations during testing. And it can sustain stable motions when subjected to uncertain human-exoskeleton interaction forces from a disabled human user. From all numerical tests performed, the effectiveness and robustness of the learned balance controller are demonstrated by its capability to maintain CoP inside the foot stable region along both lateral and forward directions.

In this study, we evaluated the controllers in a specific case for which the human musculoskeletal model has only passive muscle response (i.e., without active muscle contraction). In reality, the human-exoskeleton interaction forces might vary substantially for different users with different weights and levels of disability. One may assume that a user with good body or muscle control tends to minimize the interaction forces on the straps. On the other hand, a passive human musculoskeletal model tends to generate larger interaction forces. Therefore, using a passive musculoskeletal model can be considered a more difficult task for the exoskeleton. Further investigations with an active human musculoskeletal model are possible but will likely to need additional information on the health condition of the user.

Through dynamics randomization, our learned controller is robust against modeling inaccuracy and differences between the training and testing environments (Tan et al., 2018). Nonetheless, it is still beneficial to validate and further improve the dynamic exoskeleton model. This can be done through a model identification process, for which we can conveniently rely on the real-time measurement of GRFs and derived CoP information from the foot force sensors during programmed motions. Experiments can be conducted to correlate the CoP position with the exoskeleton posture and the CoP movement with motion. Stable squatting motions of different velocities can be used to record the dynamic responses of the exoskeleton and the collected data then can be used to further identify or optimize the model's parameters (such as inertia properties and joint friction coefficients).

By incorporating motion imitation into the learning process, the proposed robust control framework has the potential to learn a diverse array of human behaviors without the need to design sophisticated skill-specific control structure and reward functions. Common rehabilitation motions such as sit-to-stand, walking on flat or inclined ground can be learned by feeding proper target motions. Since we incorporate the reference motion as the input of the control policy network, it requires a reference motion trajectory for each new activity. The use of reference motion data (as the input of the control policy network) can alleviate the need to design sophisticated task-specific reward functions and thereby facilitates a general framework to learn a diverse array of behaviors. Note the reference motion provides guidance and needs not to be precise or even physically feasible. In this paper, the reference motion is manually generated by mimicking a human squatting, and we use it to guide the exoskeleton squatting, knowing the motion maybe not be feasible for the exoskeleton to follow exactly due to the differences in mass and inertia properties. Nonetheless, during training, the dynamic simulation environment automatically generates dynamically consistent (thus physically feasible) motions while trying to maximize the tracking reward. For other rehabilitation motions such as walking on flat or inclined ground, sit-to-stand, reference motions can be generated similarly without too much effort (unlike conventional motion or trajectory prediction or optimization methods). In addition, due to the nature of imitation learning and CoP based balance control, we foresee minimal changes to the learning framework with the exception of crafting different target motions for imitation. The learning process will automatically create specific controllers that can produce physically feasible and stable target motions (even when the target motion is coarsely generated and may not be physically feasible).

Transferring or adapting RL control policies trained in the simulations to the real hardware remains a challenge in spite of some demonstrated successes. To bridge the so-called "sim-to-real" gap, we have adopted dynamics randomization during training that is often used to prepare for sim-to-real transfer (Tan et al., 2018). In a recent work by Exarchos et al. (2020), it is shown kinematic domain randomization can also be effective for policy transfer. Additionally, efficient adaptation techniques such as latent space (Clavera et al., 2019; Yu et al., 2019) or meta-learning (Yu et al., 2020) can also be applied to further improve the performance of pre-trained policies in the real environment. We plan to construct an adaptation strategy to realize the sim-to-real control policy transfer for the lower extremity exoskeleton robot in the near future.

6 CONCLUSION

In this work, we have presented a robust, RL-based controller for exoskeleton squatting assistance with human interaction. A relatively lightweight lower extremity exoskeleton is presented and used to build a human-exoskeleton interaction model in the simulation environment for learning the robust controller. The exoskeleton foot CoP information collected from the force sensors is leveraged as a feedback for balance control and adversary perturbations and uncertain human interaction forces are used to train the controller. We have successfully demonstrated the lower extremity exoskeleton's capability to carry a human to perform squatting motions with a moderate torque requirement and provided evidence of its effectiveness and robustness. With the actuation of the ankle inversion/eversion joint, the learned controllers are also capable of maintaining the balance within the frontal plane under large perturbation or interaction forces. The success demonstrated in this study has implications for those seeking to apply reinforcement learning to control robots to imitate diverse human behaviors with strong balance and increased robustness even when subjected to the larger perturbations. Most recently, we have extended the proposed controller to perform flat ground walking with additional rewards and improved training methods and obtained stable interactive human-exoskeleton walking motions, which demonstrates the versatility of the method that will be presented in a future work. In the near future, we plan to further extend this framework to learn a family of controllers that can perform a variety of human skills like sit-to-stand, inclined ground walking and stair climbing. Lastly, these learned controllers will be deployed to the hardware through a sim-to-real transfer method and experimental tests will be performed to validate the controllers' performance.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding author.

AUTHOR CONTRIBUTIONS

In this work, XZ first proposed the research idea and approach of the paper and provided the project guidance. The code implementation, development and data analysis were done by SL and XZ. The CAD design, development and fabrication of the robotic exoskeleton were done by GJA and EN and its multibody model was created by XZ with support from GJA, EN, and SA. The first draft of the manuscript was written by SL and XZ. SA, GJA, and HS provided valuable suggestions and feedback on the draft, and HS also made some important revisions to the final paper. All authors contributed to the article and approved the submitted version.

FUNDING

This work was partially supported by the National Institute on Disability, Independent Living, and Rehabilitation Research (NIDILRR) funded Rehabilitation Engineering Research Center Grant 90RE5021-01-00, and by National Science Foundation (NSF) Major Research Instrumentation grant 1625644). HS was partially supported by the NIDILRR grant 90DPGE0011.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frobt.2021.702845/full#supplementary-material

REFERENCES

Androwis, G. J., Karunakaran, K., Nunez, E., Michael, P., Yue, G., and Foulds, R. A. (2017). Research and Development of New Generation Robotic Exoskeleton for over Ground Walking in Individuals with Mobility Disorders (Novel Design and Control). In 2017 International Symposium on Wearable Robotics and Rehabilitation (WeRob). IEEE, 1–2.

Ayas, M. S., and Altas, I. H. (2017). Fuzzy Logic Based Adaptive Admittance Control of a Redundantly Actuated Ankle Rehabilitation Robot. *Control. Eng. Pract.* 59, 44–54. doi:10.1016/j.conengprac.2016.11.015

Baraff, D. (1994). Fast Contact Force Computation for Nonpenetrating Rigid Bodies. In Proceedings of the 21st annual conference on Computer graphics and interactive techniques. 23–34.

Baud, R., Fasola, J., Vouga, T., Ijspeert, A., and Bouri, M. (2019). Bio-inspired Standing Balance Controller for a Full-Mobilization Exoskeleton. In 2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR). IEEE, 849–854. doi:10.1109/ICORR.2019.8779440

Bayon, C., Emmens, A. R., Afschrift, M., Van Wouwe, T., Keemink, A. Q. L., Van Der Kooij, H., et al. (2020). Can Momentum-Based Control Predict Human Balance Recovery Strategies? *IEEE Trans. Neural Syst. Rehabil. Eng.* 28, 2015–2024. doi:10.1109/tnsre.2020.3005455

Bionics, R. (2020). *Rex Technology*. Available at: https://www.rexbionics.com/us/product-information/ (Accessed April, 2020).

Chen, B., Ma, H., Qin, L.-Y., Gao, F., Chan, K.-M., Law, S.-W., et al. (2016). Recent Developments and Challenges of Lower Extremity Exoskeletons. *J. Orthopaedic Translation* 5, 26–37. doi:10.1016/j.jot.2015.09.007

Clavera, I., Nagabandi, A., Liu, S., Fearing, R. S., Abbeel, P., Levine, S., et al. (2019). Learning to Adapt in Dynamic, Real-World Environments through Meta-Reinforcement Learning. In International Conference on Learning Representations.

Crossley, K. M., Zhang, W.-J., Schache, A. G., Bryant, A., and Cowan, S. M. (2011). Performance on the Single-Leg Squat Task Indicates Hip Abductor Muscle Function. *Am. J. Sports Med.* 39, 866–873. doi:10.1177/0363546510395456

Delp, S. L., Anderson, F. C., Arnold, A. S., Loan, P., Habib, A., John, C. T., et al. (2007). Opensim: Open-Source Software to Create and Analyze Dynamic Simulations of Movement. *IEEE Trans. Biomed. Eng.* 54, 1940–1950. doi:10.1109/tbme.2007.901024

Emmens, A., van Asseldonk, E., Masciullo, M., Arquilla, M., Pisotta, I., Tagliamonte, N. L., et al. (2018). Improving the Standing Balance of Paraplegics through the Use of a Wearable Exoskeleton. In 2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob). 707–712. doi:10.1109/BIOROB.2018.8488066

Exarchos, I., Jiang, Y., Yu, W., and Liu, C. K. (2020). Policy Transfer via Kinematic Domain Randomization and Adaptation. arXiv preprint arXiv:2011.01891

Glorot, X., and Bengio, Y. (2010). Understanding the Difficulty of Training Deep Feedforward Neural Networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics. 249–256.

Horak, F. B., Henry, S. M., and Shumway-Cook, A. (1997). Postural Perturbations: New Insights for Treatment of Balance Disorders. *Phys. Ther.* 77, 517–533. doi:10.1093/ptj/77.5.517

Hu, J., Hou, Z., Zhang, F., Chen, Y., and Li, P. (2012). Training Strategies for a Lower Limb Rehabilitation Robot Based on Impedance Control. In 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2012. IEEE, 6032–6035. doi:10.1109/EMBC.2012.6347369

Hwangbo, J., Lee, J., Dosovitskiy, A., Bellicoso, D., Tsounis, V., Koltun, V., et al. (2019). Learning Agile and Dynamic Motor Skills for Legged Robots. *Sci. Robot* 4. doi:10.1126/scirobotics.aau5872

Karunakaran, K. K., Abbruzzese, K., Androwis, G., and Foulds, R. A. (2020). A Novel User Control for Lower Extremity Rehabilitation Exoskeletons. *Front. Robot. AI* 7, 108. doi:10.3389/frobt.2020.00108

Koolen, T., Bertrand, S., Thomas, G., De Boer, T., Wu, T., Smith, J., et al. (2016). Design of a Momentum-Based Control Framework and Application to the Humanoid Robot Atlas. *Int. J. Hum. Robot.* 13, 1650007. doi:10.1142/s0219843616500079

Kumar, V. C., Ha, S., Sawicki, G., and Liu, C. K. (2020). Learning a Control Policy for Fall Prevention on an Assistive Walking Device. In 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 4833–4840.

Lee, J., X. Grey, M., Ha, S., Kunz, T., Jain, S., Ye, Y., et al. (2018). Dart: Dynamic Animation and Robotics Toolkit. *Joss* 3 (22), 500. doi:10.21105/joss.00500

Lee, S.-H., and Goswami, A. (2012). A Momentum-Based Balance Controller for Humanoid Robots on Non-level and Non-stationary Ground. *Auton. Robot* 33, 399–414. doi:10.1007/s10514-012-9294-z

Lee, S., Park, M., Lee, K., and Lee, J. (2019). Scalable Muscle-Actuated Human Simulation and Control. *ACM Trans. Graph.* 38, 1–13. doi:10.1145/3306346.3322972

McGinty, G., Irrgang, J. J., and Pezzullo, D. (2000). Biomechanical Considerations for Rehabilitation of the Knee. *Clin. Biomech.* 15, 160–166. doi:10.1016/s0268-0033(99)00061-3

Mergner, T., and Lippi, V. (2018). Posture Control-Human-Inspired Approaches for Humanoid Robot Benchmarking: Conceptualizing Tests, Protocols and Analyses. *Front. Neurorobot.* 12, 21. doi:10.3389/fnbot.2018.00021

Mungai, M. E., and Grizzle, J. (2020). Feedback Control Design for Robust Comfortable Sit-To-Stand Motions of 3d Lower-Limb Exoskeletons. *IEEE Access.*

Nunez, E. H., Michael, P. A., and Foulds, R. (2017). 2-dof Ankle-Foot System: Implementation of Balance for Lower Extremity Exoskeletons. In 2017 International Symposium on Wearable Robotics and Rehabilitation (WeRob). IEEE, 1–2.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). Pytorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems 32. Curran Associates, Inc., 8024–8035.

Peng, X. B., Abbeel, P., Levine, S., and van de Panne, M. (2018). DeepMimic. *ACM Trans. Graph.* 37, 1–14. doi:10.1145/3197517.3201311

Peng, X. B., Berseth, G., and Van de Panne, M. (2016). Terrain-adaptive Locomotion Skills Using Deep Reinforcement Learning. *ACM Trans. Graph.* 35, 1–12. doi:10.1145/2897824.2925881

Peng, X. B., Berseth, G., Yin, K., and Van De Panne, M. (2017). DeepLoco. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 1–13. doi:10.1145/3072959.3073602

Peng, X. B., Coumans, E., Zhang, T., Lee, T.-W., Tan, J., and Levine, S. (2020). Learning Agile Robotic Locomotion Skills by Imitating Animals. arXiv preprint arXiv:2004.00784.

Rajasekaran, V., Aranda, J., Casals, A., and Pons, J. L. (2015). An Adaptive Control Strategy for Postural Stability Using a Wearable Robot. *Robotics Autonomous Syst.* 73, 16–23. doi:10.1016/j.robot.2014.11.014

Sadeghi, F., and Levine, S. (2016). Cad2rl: Real Single-Image Flight without a Single Real Image. arXiv preprint arXiv:1611.04201.

Salem, G. J., and Powers, C. M. (2001). Patellofemoral Joint Kinetics during Squatting in Collegiate Women Athletes. *Clin. Biomech.* 16, 424–430. doi:10.1016/s0268-0033(01)00017-1

Sardain, P., and Bessonnet, G. (2004). Forces Acting on a Biped Robot. center of Pressure-Zero Moment point. In IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 34, 630–637. doi:10.1109/tsmca.2004.832811

Schrade, S. O., Dätwyler, K., Stücheli, M., Studer, K., Türk, D. A., Meboldt, M., et al. (2018). Development of Varileg, an Exoskeleton with Variable Stiffness Actuation: First Results and User Evaluation from the Cybathlon 2016. *J. Neuroeng Rehabil.* 15, 18. doi:10.1186/s12984-018-0360-4

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal Policy Optimization Algorithms. arXiv preprint arXiv:1707.06347.

Shi, D., Zhang, W., Zhang, W., and Ding, X. (2019). A Review on Lower Limb Rehabilitation Exoskeleton Robots. *Chin. J. Mech. Eng.* 32, 74. doi:10.1186/s10033-019-0389-8

Tan, J., Zhang, T., Coumans, E., Iscen, A., Bai, Y., Hafner, D., et al. (2018). Sim-to-real: Learning Agile Locomotion for Quadruped Robots. arXiv preprint arXiv:1804.10332.

Thelen, D. G. (2003). Adjustment of Muscle Mechanics Model Parameters to Simulate Dynamic Contractions in Older Adults. *J. Biomech. Eng.* 125, 70–77. doi:10.1115/1.1531112

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. (2017). Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 23–30.

Vouga, T., Baud, R., Fasola, J., Bouri, M., and Bleuler, H. (2017). TWIICE - A Lightweight Lower-Limb Exoskeleton for Complete Paraplegics. In 2017 International Conference on Rehabilitation Robotics (ICORR). IEEE, 1639–1645. doi:10.1109/ICORR.2017.8009483

Wandercraft (2020). *Atalante*. Available at: https://www.wandercraft.eu/en/exo/ (Accessed April, 2020).

Xiong, M. (2014). Research on the Control System of the Lower Limb Rehabilitation Robot under the Single Degree of freedom. In Applied Mechanics and Materials. Switzerland: Trans Tech Publ, 15–20. doi:10.4028/www.scientific.net/amm.643.15

Yu, S., Huang, T.-H., Wang, D., Lynn, B., Sayd, D., Silivanov, V., et al. (2019). "Sim-to-Real Transfer For Biped Locomotion," In IEEE/RSJ International Conference on Intelligent Robots and Systems IROS, Macau, China, 4, 4579–4586. doi:10.1109/lra.2019.2931427

Yu, W., Kumar, V. C., Turk, G., and Liu, C. K. (2019). Sim-to-real Transfer for Biped Locomotion. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, 3503–3510.

Yu, W., Liu, C. K., and Turk, G. (2018). Policy Transfer with Strategy Optimization. arXiv preprint arXiv:1810.05751.

Yu, W., Tan, J., Bai, Y., Coumans, E., and Ha, S. (2020). Learning Fast Adaptation with Meta Strategy Optimization. *IEEE Robot. Autom. Lett.* 5, 2950–2957. doi:10.1109/lra.2020.2974685

Zhang, T., Tran, M., and Huang, H. (2018). Design and Experimental Verification of Hip Exoskeleton with Balance Capacities for Walking Assistance. *Ieee/asme Trans. Mechatron.* 23, 274–285. doi:10.1109/TMECH.2018.2790358

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.