IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 12, 2019, accepted July 5, 2019, date of publication July 9, 2019, date of current version July 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2927606

Terrain Adaptive Walking of Biped Neuromuscular Virtual Human Using Deep Reinforcement Learning

JIANPENG WANG, WENHU QIN, AND LIBO SUN
School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China

Corresponding authors: Jianpeng Wang (jianpeng@seu.edu.cn) and Wenhu Qin (qinwenhu@seu.edu.cn)

This work was supported in part by the National Science and Technology Major Project of the Ministry of Science and Technology of China under Grant 2014ZX07405002, in part by the Key R&D Program of Jiangsu Province under Grant BE2017035, and in part by the Fundamental Research Funds for the Central Universities under Grant 2242019k30043.

ABSTRACT There have been some biomechanics-based control systems that have achieved better realistic virtual human motion. Yet their abilities to adapt the changing environments are weaker than the traditional control systems with characters driven by proportional derivative actuators directly. In our method, we build a hierarchical neuromuscular virtual human (NMVH) motion control system that consists of a low-level spine reflex layer and a high-level policy control layer. The spine reflex layer uses a feedback net to map sensory information to excitations, which stimulate muscles to generate joint torques. The policy control layer includes a deep neural network, which provides a learned action policy to spine reflex layer for achieving terrain-adaptive motion skills. The particle swarm optimization algorithm is used to optimize the gain factors of the feedback net for finding out a basic policy to make the virtual human walk on the flat terrain autonomously. The proximal policy optimization algorithm is employed to train the deep neural network in policy control layer for learning how to modulate the actions to adapt to the changing terrain. The simulation results in Matlab show that virtual human can walk smoothly and better adapt to the given terrain changes. It demonstrates that our control system improves the terrain-adaptive walking skill of the neuromuscular virtual human.

INDEX TERMS Biped walking, hierarchical control, neuromuscular virtual human, proximal policy optimization.

I. INTRODUCTION

Virtual human plays an important role in many applications including video games, film and virtual reality. Among many approaches proposed to simulate human movement, efforts that designing the motion controllers ideally capable of adapting to the body's environment have progressed steadily. Not only the control models of virtual human are important in determining the perceived motion quality of an application, but also have the potential to elicit new controllers for legged robots and give simulation platforms for testing walking assistive devices [1], [2]. Recently, it has shown that a biomechanics based control model with only neural reflexes can generate walking close to human kinematics, dynamics and muscle stimulations [3], [4]. And this control model is already used to drive virtual human to walk and run like real humans [5]. However, when the environment changes, it is necessary to optimize all parameters again to get a new ideal result. Obviously, it is impractical that virtual human stores look-up tables of hundreds or thousands of control parameters for all different environments and behaviors.

Recently, deep reinforcement learning (DRL) provides a promising approach for developing dynamic character controllers with adaptive advantage on the irregular terrains. Furthermore, these controllers trained with DRL can generate motions with high quality like state-of-the-art kinematic methods [6]. As with most physics based control methods, characters in these DRL trained control frameworks are still driven by proportional-derivative (PD) controllers that produce the torques applied to the joints according to specified target angles. However, these torques acquired from PD controllers are hardly expressing for the truly actuate procedure in a real human motion, also for the forces interaction between human and environment. Therefore, aiming at strengthen terrain-adaptive walking skills, a hierarchical framework with a deep neural network is integrated into

The associate editor coordinating the review of this manuscript and approving it for publication was Lei Wei.

the biomechanics based control system. And, a deep reinforcement learning framework is provided to train control policies for improving the terrain-adaptive motion skills of the simulated biomechanics based character.

The rest of this paper is organized as follows: A quick overview of the related works is presented in Section 2. Section 3 describes the hierarchical organization of the virtual human motion control model and proposes a method of motion generation through deep reinforcement learning. The applications of our approach and the experimental results with discussions are presented in Section 4. Section 5 draws conclusions and future works.

II. RELATED WORK

Virtual humans have been popularly employed in various scenarios and must act and react in their simulated environment, and the motion control and simulation of physics-based virtual human has been a topic of interest in computer animation, robotics, and biomechanics for several decades. Physics-based motion control approach allows all motion to be the result of a physics simulation process. As a consequence, virtual human automatically interacts in a way that is physically accurate, without the need for additional motion data or scripts [7], [8]. In physically based virtual human simulation, much of the simulation techniques have been based on articulated skeleton with joint torques generated by PD controllers [8], [9]. However, these approaches do not consider the fact that the joint torques that produce motion are the direct result of muscle forces acting on these joints. Recently, the simulation of muscle-forced movements is an active research topic of growing attention, because the generated locomotion is human-like in terms of dynamics, adaptivity, and robustness [10]–[12]. Also, the muscle force driven controllers can be used to obtain motion data from the virtual human designed based on deformities or injuries, since obtaining real-world data of such characters can be a challenge. In biomechanical movement simulations, virtual humans are able to control joint torques thanks to antagonistically acting muscle pairs with the Hill-based muscle model to generate force almost exclusively [13]. Delp et al. developed a model of the human lower extremity to study how surgical changes in musculoskeletal geometry and musculotendon parameters affect muscle force and its moment about the joints [14]. However, this model contains too many muscles for a virtual human, it is usually used in medical field but not in motion simulation [15]. The neural networks along the spinal cord play a significant role in actuating muscles to generate motion behaviors, which has been proved by experiments in human and animals [16]–[18]. Inspired by neuromuscular models and reflex control, Geyer et al. demonstrated that reflex-based motor control can generate efficient and reliable bouncing gaits instead of using central motor commands [19]. Then a simple muscle-actuate bipedal walking model is designed based on a reflex loop which simulate the neural reflex network from real human and produced stable locomotion with characteristics close to real humans [4]. This neuromuscular model of Geyer and Herr is a well-accepted human walking model, which is extended to 3D and received a good locomotion result [4]. Other studies on reflex control show the important potential of this bioinspired method for understanding human motor control [20]. While significant progress has been made in recent years, motion control of muscle-forced characters remains a weak interactive ability.

The reinforcement learning (RL) offers a promise of being able to learn control strategies in a principled way. It has been applied in a number of ways to the control of walking [21]. Some RL approaches have been explored to develop more general controllers for simulated characters [22]–[24]. However, the high-dimensionality of the state spaces involved in the control of locomotion remains problematic, as does the need to design an appropriate reward function. Recently, significant progress has been made by combining advances in deep learning for learning feature representations with reinforcement learning. Notable examples are training agents to play Atari games based on raw pixels [25], [26]. Impressive results have also been obtained in training deep neural network policies for 3D locomotion and manipulation tasks [24], [27]. The DRL can be viewed as introducing deep neural networks (DNNs) to RL. Recently, it has demonstrated increasing capabilities for continuous control problems, including agents that can move with skill and agility through their environment [6], [28]. These RL-based techniques have the advantage that, compared to the gait controllers previously described, less user input is needed to hand tune the controllers, and they are more flexible to learning additional, novel tasks.

Our work builds upon relevant technical advances in deep reinforcement learning, biomechanics, and neuroscience to model the biomechanical characteristics of the human body and to emulate its motor control mechanisms. The hierarchical virtual human control system that we created includes a neural reflex layer and a policy control layer with a DNN. We adopt the proximal policy optimization (PPO) method to train the neural network in policy control layer to adjust some parameters for improving the terrain-adaptive walking skill of the neuromuscular virtual human (NMVH). In our approach, we actuate each joint by muscle tendon units (MTUs) rather than using PD controllers. This is advantageous because it allows us to properly model the complexity of the human tendon network, which we believe is important for obtaining realistic motions. We conjectured that deep reinforcement learning methods would yield more realistic results with biologically accurate models and actuators. And, in this method the search space is reduced by driving the virtual human in accordance with a hierarchical control structure.

III. CONTROL SYSTEM

A schematic view of the simulation system as shown in Fig. 1. The simulation system drives the musculoskeletal human model by integrating a hierarchical control model with a high-level policy control layer and a low-level neural reflex

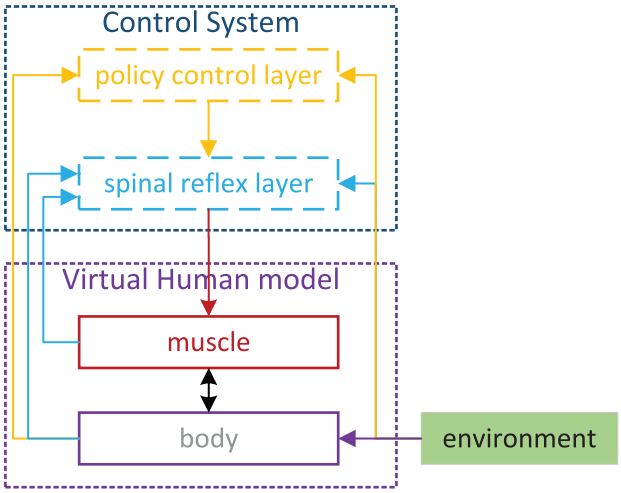


FIGURE 1. Schematic view of the NMVH motion control system.

layer. In the virtual human model, the MTUs are employed with the purpose of receiving activations from neural reflex layer and generating the muscle force to actuate the human musculoskeletal body. At neural reflex layer, a neural nerve network senses four kinds of feedback information and maps them to muscle excitations. The policy control layer has a deep neural network which is trained through the PPO method, and maps the surrounding terrain and current state of the agent to an action to adjust the basic policy for improving its terrain-adaptive motion skill.

A. MUSCULOSKELETAL MODEL

The virtual human is modeled as a three dimensional system consisted of a hierarchy of rigid bodies with seven segments connected by eight internal degrees of freedom (DOFs) and actuated by 20 MTUs, as shown in Fig. 2. The eight internal DOFs include one for each ankle and knee and two for each hip joint of both limbs. All joints are actuated by the established dynamic MTUs [4], [5]. The gluteus (GLU) and the hip flexor (HFL) permit respectively extension and flexion of the hip joint in sagittal plane. The hip abductors (HAB) and adductors (HAD) actuate the hip joint roll in coronal plane.

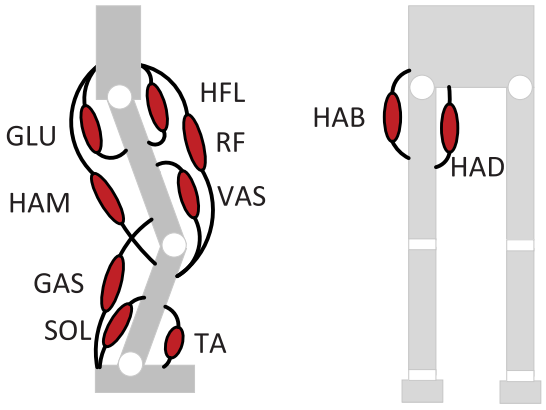


FIGURE 2. The NMVH's Musculoskeletal model.

The knee joint is extended by the Vasti (VAS). The ankle joint is driven by the gastrocnemius (GAS) and the tibialis (TA), permitting extension and flexion of the joints respectively. The three bi-articular muscle tendon units, the hamstring (HAM), the rectus femoris (RF) and the soleus (SOL), can actuate two joints at the same time. The weight and the length of different segments are based on anthropometric data from [29].

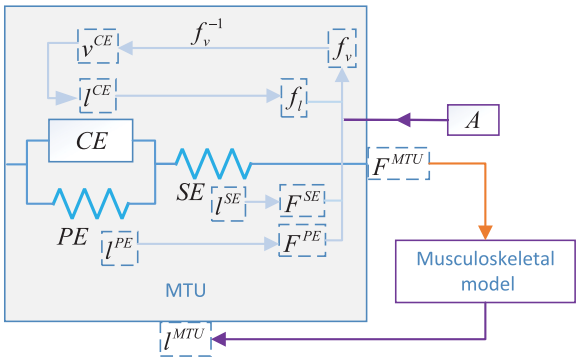


FIGURE 3. The interplay of a MTU's internal components and the relationship among the MTU's received activation A, muscle length l^{MTU} and muscle force F^{MTU}.

Fig. 3 depicts the internal composition and interplay of a MTU model and how it interacts with the neural reflex layer and the musculoskeletal model. Inside the Hill-type MTU model, the contractile element (CE) models muscle fibers that can actively generate force, the parallel-elastic element (PE) models the passive force muscle fibers in parallel with the CE, and both connect to the serial elastic element (SE) which models the tendon. The total force of the model is equal to the force coming from the tendon (F^{SE}) and will always equal the contribution from the muscle (F^{CE} and F^{PE}):

$$F^{\mathrm{MTU}} = F^{\mathrm{SE}} = F^{\mathrm{CE}} + F^{\mathrm{PE}}. \tag{1}$$

The active force of the CE depends on its activity $\mathcal{A}$, on its length through the f_l function and on its velocity through the f_v function:

$$F^{\mathrm{CE}} = \mathcal{A} \cdot F^{\max} \cdot f_l(l^{\mathrm{CE}}) \cdot f_v(v^{\mathrm{CE}}), \tag{2}$$

with:

$$\iota \frac{d\mathcal{A}}{dt} = ((S(t) - \mathcal{A}),$$

$$f_l(l^{\mathrm{CE}}) = exp(c|\frac{l^{\mathrm{CE}} - l^{\mathrm{opt}}}{l^{\mathrm{opt}} w}|^3),$$

$$f_v(v^{\mathrm{CE}}) = \begin{cases} \dfrac{v^{\max} - v^{\mathrm{CE}}}{v^{\max} + Kv^{\mathrm{CE}}}, & \text{if } v^{\mathrm{CE}} < 0 \\ N + (N-1)\dfrac{v^{\max} + v^{\mathrm{CE}}}{7.56Kv^{\mathrm{CE}} - v^{\max}}, & \text{if } v^{\mathrm{CE}} \geqslant 0. \end{cases}$$

The activity is modeled as a first order differential equation of the stimulation S sent to the muscle, where ι is a time constant. In the force-length relationship, l^{opt} is the optimum CE length for maximum force production, w describes the width of the

$f_l(l^{CE})$ curve and c is ln(0.05). In the force-velocity relationship, $v^{max} < 0$ is the maximum contraction velocity, K is a curvature constant and N is the dimensionless amount of force F^{MTU}/F^{max} reached at a lengthening velocity $v^{CE} - v^{max}$. In addition, given the tendon slack length l^{slack}, the passive force F^{PE} acts against muscle movement when the muscle is slack (i.e. if $l^{MTU} - l^{CE} > l^{slack}$) or stretches beyond its optimal length (i.e. if $l^{CE} > l^{opt}$):

$$F^{PE} = \begin{cases} F^{max}(\frac{l^{opt}-w-l^{CE}}{l^{opt}w/2})^2, & \text{if } l^{MTU} - l^{CE} > l^{slack} \\ F^{max}(\frac{l^{CE}-l^{opt}}{l^{opt}w})^2 f_v(v^{CE}), & \text{if } l^{CE} > l^{opt} \\ 0, & \text{else} \end{cases}. \tag{3}$$

The tendon is active only if it extends beyond its slack length ($l^{SE} > l^{slack}$) and its force is modeled as a square function of the normalized tendon length $\epsilon(l^{SE}) = \frac{l^{SE}-l^{slack}}{l^{slack}}$:

$$F^{SE} = \begin{cases} F^{max}(\frac{\epsilon}{\epsilon^{ref}})^2, & \text{if } \epsilon > 0 \\ 0, & \text{else} \end{cases}. \tag{4}$$

When the MTU force F^{MTU} is computed for a given muscle activity $\mathcal{A}$ and MTU length l^{MTU}, it is used to deduce the knee or ankle joint torque: $\tau = r_j(\theta - \theta^{max})F^{MTU}$. Where θ is the current joint angle and r_j is the maximum MTU-joint moment arm with $\theta = \theta^{max}$. MTUs attached to the hip are assumed to have a constant moment arm: $\tau = r_j F^{MTU}$.

B. NEURAL REFLEX LAYER

The muscle force variation of a MTU is relative to the stimulations it receives. The neural reflex layer models the spinal reflex process of human which receive command and sensory information and send stimulations to muscles. As shown in Fig. 4, the neural reflex layer contains a feedback net which receives control signals from policy control layer, senses four kinds of feedback information from the body and the environment and maps all inputs to muscle excitations. The time delay between policy control layer and neural reflex net layer is set to 5ms. The neural transmission delay for hip, knee and ankle to the neural reflex network are set to 2.5ms, 5ms and 10ms respectively [30]–[32].

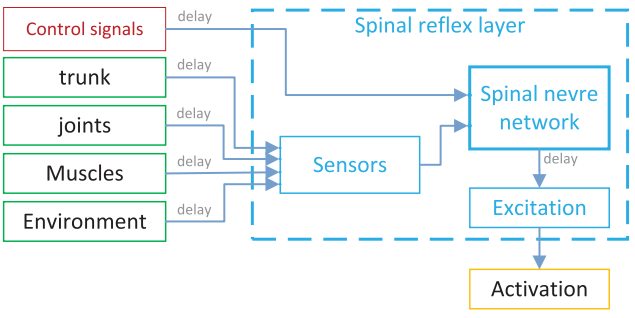


FIGURE 4. Neural reflex layer receive control signals and sensory information and map them as the stimulations to MTUs.

There are three types of muscle sensory information from all MTUs, including the length feedback that models the muscle spindle, force feedback that models the Golgi tendon, and excitation feedback that is used to reflect the correlation between muscles. The muscle length feedback from a specific MTU M is defined as:

$$B_M^L = l_M^{CE}/l_M^{opt}(t - \Delta t_M) - l_M^{offset}, \tag{5}$$

where, l_M^{CE}, l_M^{opt} are respectively the CE length and the CE optimal length of MTU M, l_M^{offset} is a positive parameter found by optimization and Δt_M is the delay of information sensing. The muscle force feedback equation for a given MTU M is defined as:

$$B_M^F = F_M^{MTU}/F_M^{max}(t - \Delta t_M), \tag{6}$$

where, F_M^{MTU} and F_M^{max} are respectively corresponding to the current force generated by M and the maximum force that can be generated. The muscle excitation feedback equation is defined as:

$$B_M^S = S_M^S(t - \Delta t_M) \tag{7}$$

with S_S^M being the excitation of the muscle M from contralateral leg.

The joint sensory information is used to prevent joint overextension. When the joint angle exceeds the limit joint angle, the joint angle feed for preventing the joint overextension:

$$B_J = \begin{cases} (\phi_j - \hat{\phi}_j)(t - \Delta t_j), & \text{if } \Delta\phi > 0, \omega/\omega^{ref} > -1 \\ 0, & \text{else,} \end{cases} \tag{8}$$

where, Δt_j is the time-delay, ϕ_j is the actual angle of joint j, $\hat{\phi}_j$ is the max tolerated angle, ω is the angular speed and ω^{ref} is the reference angular speed which is used to normalize the joint angular speed. The sign of ω is positive when it is going toward the joint limit angle.

The mainly environment sensory information is the contact force between the ground and the foot. The total ground feedback from one foot is defined as the sum of the normalized output from all contact points:

$$B_G = k_G \frac{\sum F_{pt}(t - \Delta t_{pt})}{m \cdot g}, \tag{9}$$

where, g is the gravity, m is the mass of the body, Δt_{pt} is the time-delay, F_{pt} is the output force of contact point and k_G is a factor found by optimization.

The torso sensory information is the current state of the torso. The torso feedback function is defined as a PD control law that aims to bring the actual torso angle θ toward a reference angle $\hat{\theta}$:

$$B_T = \{k_p(\theta(t - \Delta t_T) - \hat{\theta}) + k_d\dot{\theta}(t - \Delta t_T)\}_{\pm}, \tag{10}$$

where, k_p and k_d are control parameters from policy control layer, Δt_T is the time-delay that neural reflex net receive sensory information from torso. The sign of the brackets depends on the action of the muscles on the trunk; Negative

if the action of the muscle is in the direction of positive angle changes and positive otherwise [5].

For each MTU, the stimulation S_M that it received from neural reflex layer is modeled as a basal stimulation S_M^0 plus a linear combination of the weighted feedback sensory information:

$$S_M = \begin{cases} \widehat{S}_M + \sum G_S^M B_S, & \text{if } 0 \leqslant S_M \leqslant 1 \\ 0, & \text{if } S_M < 0 \\ 1, & \text{if } S_M > 1, \end{cases} \tag{11}$$

where, $\widehat{S}_M$ is the basal excitation for M, B_S is the feedback value from one sensor which has an impact on M, and G_S^M is a positive gain factor of the map.

Two different phases of walking were defined as submodels: the stance phase and the swing phase [33]. The stance phase is modeled began at the exact instant that the foot made contact with the ground. In this model, the main task of the MTUs on the support leg is to push the body forward and simultaneously keep the trunk balanced. The relationship between feedback sensory information and muscle excitations in stance phase is defined as:

$$\begin{cases} S_{\text{HAB}}^{\text{stance}} = \widehat{S}_{\text{HAB}}^{\text{stance}} + G_{\text{HAB}_{\text{F}}}^{\text{HAB}} B_{\text{F}}^{\text{HAB}} + G_{\text{T}}^{\text{HAB}} B_{\text{T}}^{\text{coronal}} \\ \qquad + G_{\text{HAB}_{\text{S}}^{\text{opp}}}^{\text{HAB}} B_{\text{S}}^{\text{HAB}^{\text{OPP}}} \\ S_{\text{HAD}}^{\text{stance}} = \widehat{S}_{\text{HAD}}^{\text{stance}} + G_{\text{T}}^{\text{HAD}} B_{\text{T}}^{\text{coronal}} + G_{\text{HAD}_{\text{S}}^{\text{opp}}}^{\text{HAD}} B_{\text{S}}^{\text{HAD}^{\text{OPP}}} \\ S_{\text{GLU}}^{\text{stance}} = \widehat{S}_{\text{GLU}}^{\text{stance}} + G_{\text{GLU}_{\text{F}}}^{\text{GLU}} B_{\text{F}}^{\text{GLU}} + G_{\text{T}}^{\text{GLU}} B_{\text{T}}^{\text{sagittal}} \\ \qquad + G_{\text{HFL}_{\text{S}}^{\text{opp}}}^{\text{GLU}} B_{\text{S}}^{\text{HFL}^{\text{OPP}}} \\ S_{\text{HFL}}^{\text{stance}} = \widehat{S}_{\text{HFL}}^{\text{stance}} + G_{\text{HFL}_{\text{L}}}^{\text{HFL}} B_{\text{L}}^{\text{HFL}} + G_{\text{T}}^{\text{HFL}} B_{\text{T}}^{\text{sagittal}} \\ \qquad + G_{\text{GLU}_{\text{S}}^{\text{opp}}}^{\text{HFL}} B_{\text{S}}^{\text{GLU}^{\text{OPP}}} + G_{\text{HAM}_{\text{S}}^{\text{opp}}}^{\text{HFL}} B_{\text{S}}^{\text{HAM}^{\text{OPP}}} \\ S_{\text{HAM}}^{\text{stance}} = \widehat{S}_{\text{HAM}}^{\text{stance}} + G_{\text{HAM}_{\text{F}}}^{\text{HAM}} B_{\text{F}}^{\text{HAM}} + G_{\text{GLU}_{\text{S}}}^{\text{HAM}} B_{\text{S}}^{\text{GLU}} \\ S_{\text{VAS}}^{\text{stance}} = \widehat{S}_{\text{VAS}}^{\text{stance}} + G_{\text{VAS}_{\text{F}}}^{\text{VAS}} B_{\text{F}}^{\text{VAS}} - G_{\text{knee}} B_{\text{knee}} \\ S_{\text{SOL}}^{\text{stance}} = \widehat{S}_{\text{SOL}}^{\text{stance}} + G_{\text{SOL}_{\text{F}}}^{\text{SOL}} B_{\text{F}}^{\text{SOL}} \\ S_{\text{TA}}^{\text{stance}} = \widehat{S}_{\text{TA}}^{\text{stance}} + G_{\text{TA}_{\text{L}}}^{\text{TA}} B_{\text{L}}^{\text{TA}} + G_{\text{TA}_{\text{F}}}^{\text{TA}} B_{\text{F}}^{\text{SOL}} \\ S_{\text{GAS}}^{\text{stance}} = \widehat{S}_{\text{GAS}}^{\text{stance}} + G_{\text{GAS}_{\text{F}}}^{\text{GAS}} B_{\text{F}}^{\text{GAS}} \end{cases} \tag{12}$$

where, S_M^{stance} is the stimulation that the neural reflex network output to MTU M in stance phase, $\widehat{S}_M^{\text{stance}}$ is its basal stimulation. B_F^M and B_L^M are the force and length feedback information from MTU M. B_T^{sagittal} and B_T^{coronal} are the trunk feedback information from sagittal plane and coronal plane. $B_S^{M^{\text{OPP}}}$ is the excitation feedback information and used to compensate for the moment induced on the trunk by the contralateral swing leg. B_{knee} is the joint angle feedback information of knee. The feedback gain G is found by optimization.

Swing Phase began at the point where the foot left the ground. In this phase, MTUs on the swing leg works for placing the leg into target angles in sagittal and coronal planes. The relationship between feedback sensory information and muscle excitations in swing phase is defined as:

$$\begin{cases} S_{\text{HAB}}^{\text{swing}} = \widehat{S}_{\text{HAB}}^{\text{swing}} + G_{\text{HAB}_{\text{L}}}^{\text{HAB}} B_{\text{L}}^{\text{HAB}} + G_{\delta}^{\text{HAB}} S_{\delta}^{\text{coronal}} \\ S_{\text{HAD}}^{\text{swing}} = \widehat{S}_{\text{HAD}}^{\text{swing}} + G_{\text{HAD}_{\text{L}}}^{\text{HAD}} B_{\text{L}}^{\text{HAD}} + G_{\delta}^{\text{HAD}} S_{-\delta}^{\text{coronal}} \\ S_{\text{GLU}}^{\text{swing}} = \widehat{S}_{\text{GLU}}^{\text{swing}} + G_{\text{GLU}_{\text{L}}}^{\text{GLU}} B_{\text{L}}^{\text{GLU}} + G_{\delta}^{\text{GLU}} S_{-\delta}^{\text{sagittal}} \\ S_{\text{HFL}}^{\text{swing}} = \widehat{S}_{\text{HFL}}^{\text{swing}} + G_{\text{HFL}_{\text{L}}}^{\text{HFL}} B_{\text{L}}^{\text{HFL}} + G_{\delta}^{\text{HFL}} S_{\delta}^{\text{sagittal}} \\ S_{\text{HAM}}^{\text{swing}} = \widehat{S}_{\text{HAM}}^{\text{swing}} + G_{\text{HAM}_{\text{L}}}^{\text{HAM}} B_{\text{L}}^{\text{HAM}} + G_{\delta}^{\text{HAM}} B_{-\delta}^{\text{GLU}} \\ S_{\text{VAS}}^{\text{swing}} = \widehat{S}_{\text{VAS}}^{\text{swing}} + G_{\text{VAS}_{\text{L}}}^{\text{VAS}} B_{\text{L}}^{\text{VAS}} \\ S_{\text{SOL}}^{\text{swing}} = \widehat{S}_{\text{SOL}}^{\text{swing}} + G_{\text{GAS}_{\text{S}}}^{\text{SOL}} B_{\text{S}}^{\text{GAS}} \\ S_{\text{TA}}^{\text{swing}} = \widehat{S}_{\text{TA}}^{\text{swing}} + G_{\text{TA}_{\text{L}}}^{\text{TA}} B_{\text{L}}^{\text{TA}} + G_{\text{SOL}_{\text{F}}}^{\text{TA}} B_{\text{F}}^{\text{SOL}} \\ S_{\text{GAS}}^{\text{swing}} = \widehat{S}_{\text{GAS}}^{\text{swing}} + G_{\text{HAM}_{\text{F}}}^{\text{GAS}} B_{\text{F}}^{\text{HAM}} \end{cases} \tag{13}$$

In addition, if a leg switch from stance phase to swing phase during the transitional double support phase, the stimulations for stance control are inhibited and the stimulations for swing control are excited in proportion to contact force of contralateral leg.

C. POLICY CONTROL LAYER

The policy control layer monitors the motion of the human model in real-time and updates control policy at the moment that the foot heel of swing leg contacts ground. As shown in Fig. 5, it employs the basic policy to maintain an autonomic motion of the model if no changes in the environment, and adopts a DNN to modulate the control policy when the character observed any terrain changes. The state feature descriptions of virtual human and terrain serves as the input for the network. The output of the network is the action, a, which governs the evolution of the motion during the following steps.

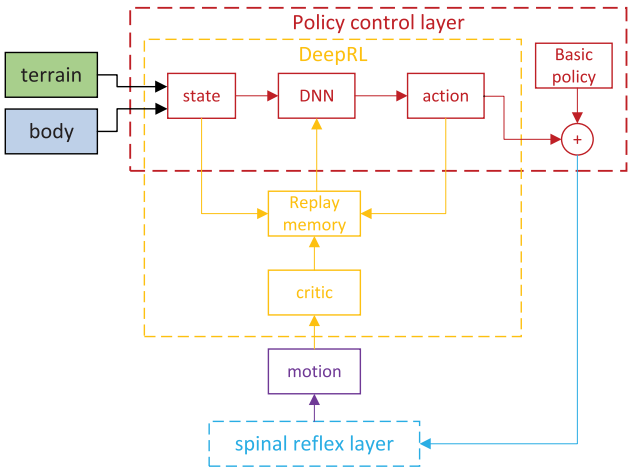


FIGURE 5. According to the state of terrain and character body pose, policy control layer determine an action for adjusting the control policy to adapt the terrain changes.

1) STATE AND ACTION

The state s, as the input of the DNN, consists of features describing the configuration of the character and the upcoming terrain height. The projection of the midpoint of two hip

joints onto the horizontal plane, zero point (ZP), is defined as the reference point for all virtual human features. All heights and positions of terrain or character are expressed relative to ZP. We set up the virtual human's perception range as about 3 biped cycles which is from (ZP-1)m to (ZP+4)m. The terrain features, T, consist of a 1D array of samples from the terrain height-field within the perception range. The samples are spaced 5cm apart, for a total of 100 height samples. The character features, C, consist of the pose p and velocity v, where p records the center of mass heights of torso, each joint, foot-heel and foot-ball and the projection positions of them on the horizontal plane, and v records the velocity of each joint. Combined, the final state representation is 154 dimensional. Fig. 6 illustrate the character and terrain features.

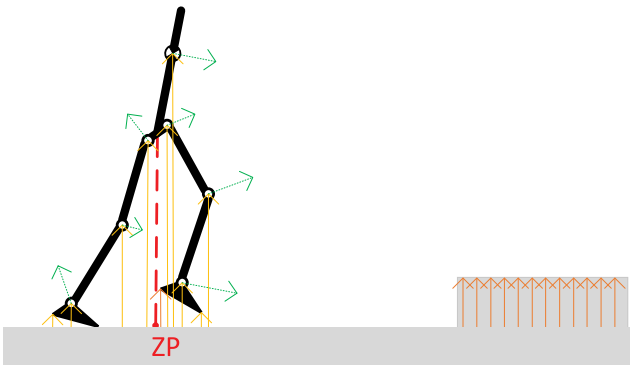


FIGURE 6. The state features of character and terrain.

The virtual human control model is guided by the segment dynamics of a double pendulum [10]. The control process can be broken down into three phases: stance phase, transitional double support phase and swing phase. During the transitional phase, the control for acting stance is inhibited and the control for swing leg is excited at the same time. Due to neural reflex network involve more reflex parameters, there are relatively many control parameters in the neuromuscular virtual human motion control system. If the action contains all parameters, it may come into being a huge action space with high learning cost. In order to meet the total system performance's need and reduce the cost of learning, only a few target parameters are taken up as the elements of the action based on the analysis of the control system. It has been stated that one of the major tasks of neural reflex layer in stance phase is to realize trunk balance by activating the hip antagonists in the sagittal and coronal planes. Therefore, two trunk balance feedback parameters in each plane are selected as the elements of the output action of the policy control DNN. In addition, the target trunk angle in sagittal plane is also selected as one element of action. A swing process that can be broken into three stages: flexing the leg to the target clearance length, advancing the leg to the target angles in sagittal and coronal planes, and extending the leg until ground contact. The target clearance length and target swing angle are two elements of the action. In addition, three parameters in each plane are included as the action elements for computing foot target placement. Thus, a total of 13 controller parameters serve to define the available policy actions.

2) STRUCTURE OF DEEP NEURAL NETWORK

A schematic diagram of the DNN is available in Fig. 7. The DNN receives a state $s = (T, B)$ as input, and first processes the terrain features T by 128 fully connected units. The resulting feature vector is then concatenated with B, and processed by two fully connected layers with 512 and 128 units. The linear output layer with 13 units produces the final action. Rectified linear units are used for all layers, except for the output layers.

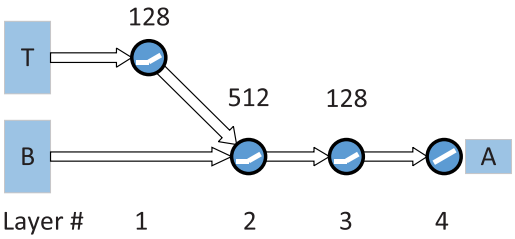


FIGURE 7. Schematic illustration of the DNN.

D. LEARNING CONTROL POLICY

We implement the simulation in the simulink environment (Matlab R2017a) with the ode15s solver. The contact forces of the model with the ground are managed by the physical simulator of Simulink. The basic policy of the virtual human control model is that the model adopts fixed optimal parameters to control walking behavior without perceiving ground changes. It aims to generate an automatic, stable and realistic virtual human walking on flat terrain without intervention of the DNN in the virtual human control model. The parameters of the basic policy are obtained after optimization by particle swarm optimization (PSO) algorithm. Once the virtual human senses a local terrain change, the DNN in policy control layer will intervene in the motion control process to adjust some parameters which are not directly related to the muscle excitation. Furthermore, the DNN is trained through PPO algorithm based on the parameters under basic policy.

1) OPTIMIZING FOR BASIC POLICY

There are 64 parameters in total for the basic policy to optimize. Most of them come from neural reflex layer, and only a few target-relevant parameters such as trunk pitch angle, swing leg angel and length are directly assigned by the basic policy [4]. The PSO is adopted to find out optimal control parameters of the basic policy [34]. The update equations for the position and velocity in the PSO algorithm are shown as:

$$x_i(n+1) = x_i(n) + v_i(n+1), \tag{14}$$

$$v_i(n+1) = \omega * v_i(n) + c_1 * (x_g - x(n)) * Rand[0:1] + c_2 * (x_p - x(t)) * Rand[0:1] \tag{15}$$

where, x_p and x_g are the personal and global best position respectively. ω is the inertia weight. c_1 is the social learning

factor and represents the attraction that a particle has toward the success of the entire swarm. c_2 is the cognitive learning factor and represents the attraction that a particle has toward its own success. The objective function is defined as:

$$R_0 = \begin{cases} 0, & \text{if charactor falls down} \\ e^{E_v+E_p+E_e}, & \text{else,} \end{cases} \tag{16}$$

where, a fall is defined as the character stumbled over an obstacle or the torso's pitch exceed the given threshold. The $E_v = -w^v \sum (v^* - v)^2$, $E_p = -w^p|p|$, and $E_e = -w^e \sum \Delta \dot{E}$ correspond to rewards (or penalties) for velocity, position, and effort respectively, and v is the average horizontal velocity of the center of mass during a cycle, v^* is the desired velocity, p is the final crosswise position deviation of the center of mass, $\sum \Delta \dot{E}$ is the total metabolic energy expenditure over all muscles [5].

2) DEEP REINFORCEMENT LEARNING FOR CONTROL POLICY

The control policies which come from policy control layer are trained with PPO, which has demonstrated state-of-the-art results on a number of challenging control problems [35]. Algorithm 1 illustrates the overall learning process. The policy control layer interacts with the environment at the beginning of each biped cycle. It receives an observation s, takes an action a, receives a scalar reward r and results in a successor s'. The reward r is defined as:

$$r(s, a, s') = \begin{cases} 0, & \text{if charactor falls down} \\ e^{E_v+E_j}, & \text{else,} \end{cases} \tag{17}$$

where, E_v is same as in basic policy, and $E_j = -w^j \sum_j \tau$ is the total penalty for joint overextension. The reward is used as a training signal to encourage the character to travel forward at a desired speed with fewer joint overextension and without falling. If the character falls during a biped cycle, it is reset to a default state and the terrain is regenerated randomly.

The learning adopts an off-policy exploration in which the deterministic policy $\pi(s) : S \rightarrow A$ maps a state $s \in S$ to an action $a \in A$, while a stochastic policy $\pi(s, a) : S \times A \rightarrow \mathbb{R}$ represents a Gaussian conditional probability distribution of a given s. For the stochastic policy, a new action $a = \mu(s) + \mathcal{N}$ can be generated by applying Gaussian noise $\mathcal{N}$ to the mean action $\mu(s)$ which is the output of the DNN with parameter κ_μ. The return from a state is defined as the sum of discounted future reward:

$$R_n = \sum_{i=n}^{N} \gamma^{(i-n)} r(s_i, a_i), \tag{18}$$

where, $\gamma \in [0 \quad 1]$ is a discounting factor. And the summation can be rewritten recursively as:

$$\begin{aligned} R_n &= r(s_n, a_n) + \gamma \sum_{i=n+1}^{N} \gamma^{i-(n+1)} r(s_i, a_i) \\ &= r(s_n, a_n) + \gamma R_{n+1} \end{aligned} \tag{19}$$

Algorithm 1 Proximal Policy Optimization

$\kappa_\mu \leftarrow$ random weights
$\kappa_v \leftarrow$ random weights
while not done **do**
  while there are not enough new tuples **do**
    $s_0 \leftarrow$ using basic policy and walking on level ground
    for step $= 1, \ldots, n$ **do**
      $s \leftarrow$ start state
      $a \leftarrow \pi(s|\kappa_\mu)$
      Apply a and simulate forward one step
      $s' \leftarrow$ next state
      $r \leftarrow$ reward
    end for
    for each step **do**
      $R_i^\lambda \leftarrow (1-\lambda) \sum_{n=1}^{\infty} \lambda^{n-1} R_i^{(n)}$
      $\hat{G}_i^\lambda \leftarrow \hat{A}_t + Q(s_i)$
      $\Gamma_i \leftarrow (s_i, a_i, s_i', R_i^\lambda, \hat{G}_i^\lambda)$
      Store Γ_i in D
    end for
  end while

  $\kappa_\mu \leftarrow \kappa_\mu^{old}$
  for each update step **do**
    Sample minibatch of n samples Γ_i from D

    Update value function :
    for each Γ_i **do**
      $Q(s_i) \leftarrow$ critic s_i
    end for
    $\kappa_v \leftarrow \kappa_v + \alpha_v \left(\frac{1}{n} \sum_i \nabla_{\kappa_v} Q(s_i|\kappa_v)(R_i^\lambda - Q(s_i)) \right)$

    Update policy :
    for each Γ_i **do**
      $Q(s_i) \leftarrow$ critic s_i
      $\hat{A}_i \leftarrow \hat{G}_i^\lambda - Q(s_i)$
      $\rho_i(\kappa_\mu) = \frac{\pi(a_i, s_i|\kappa_\mu)}{\pi(a_i, s_i|\kappa_\mu^{old})}$
    end for
    $\kappa_\mu \leftarrow$
    $\kappa_\mu + \alpha_\pi \frac{1}{n} \sum_i \nabla_{\kappa_\mu} \min(\rho_i(\kappa_\mu)\hat{A}_i, \text{clip}(\rho_i(\kappa_\mu), 1-\epsilon, 1+\epsilon)\hat{A}_i)$
  end for
end while

The goal of reinforcement learning is to find a control policy which maximizes the expected returns from the start state $\mathbb{E}_\pi[R_1]$. The value function is used to describe the expected return after taking an action a_n in state s_n and thereafter following policy π : $Q^\pi(s_n, a_n) = \mathbb{E}[R_n|s_n, a_n]$.

The policy $\pi(s, a|\kappa_\mu)$ and the value function $Q(s|\kappa_v)$ are learned in tandem during the PPO-based reinforcement learning process. The value function is updated using the temporal difference computed with the λ-return results in the TD(λ) algorithm [36]. The n-step return can be computed by

truncating the sum of returns after n steps:

$$R_t^{(n)} = \sum_{l=0}^{n-1} \gamma^l r_{t+l} + \gamma^n Q(s_{t+n}), \tag{20}$$

where, $Q(s)$ is the value function which approximates the return from the remaining steps. And on this return and a decay factor λ the λ-return can be obtain by:

$$R_t^{\lambda} = (1-\lambda) \sum_{n=1}^{\infty} \lambda^{n-1} R_t^{(n)}. \tag{21}$$

Similarly, the policy is updated using gradients computed from the surrogate objective, with advantages $\hat{A}_t$ computed using generalized advantage estimator GAE(λ) [24]:

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \cdots + \cdots + (\gamma\lambda)^{T-t+1}\delta_{T-1}, \tag{22}$$

where, $\delta_t = r_t + \gamma Q(s_{t+1}) - Q(s_t)$, t specifies the time index in [0, T], within a given length T trajectory segment. The GAE(λ) for current state can be computed through: $\hat{G}_t^{\lambda} = \hat{A}_t + Q(s_t)$.

The character's experiences are summarized by transition tuples $\Gamma = (s, a, s', R^{\lambda}, \hat{G}^{\lambda})$. A replay buffer D with size of 40k is used to sample and store the tuples of each biped cycle. When the replay buffer was full, the oldest samples were discarded. During a value update, a mini-batch of tuples $\{\Gamma_i\}$ are sampled uniformly from the replay buffer D. After sampling a minibatch data from the replay buffer D, the value function is updated using stochastic gradient descent (SGD) method with target values:

$$\kappa_v \leftarrow \kappa_v + \alpha_v\Big(\frac{1}{n}\sum_i \nabla_{\kappa_v} Q(s_i|\kappa_v)(R_i^{\lambda} - Q(s_i))\Big), \tag{23}$$

where, α_v is the critic learning rate. Let $\rho_t(\kappa_\mu)$ donotes the probability ratio $\rho_t(\kappa_\mu) = \frac{\pi(a_t, s_t|\kappa_\mu)}{\pi(a_t, s_t|\kappa_\mu^{old})}$. The clipped surrogate loss $L^{CLIP}(\kappa_\mu)$ is defined as:

$$L^{CLIP}(\kappa_\mu) = \mathbb{E}[\min(\rho_t(\kappa_\mu)\hat{A}_t, \mathrm{clip}(\rho_t(\kappa_\mu), 1-\epsilon, 1+\epsilon)\hat{A}_t)|s_t, a_t], \tag{24}$$

where, ϵ is a hyperparameter. The clip modifies the surrogate objective by clipping the probability ratio, which removes the incentive for moving ρ_t outside of the interval $[1-\epsilon, 1+\epsilon]$. therefore, the policy can be updated using SGD:

$$\kappa_\mu \leftarrow \kappa_\mu + \alpha_\pi \frac{1}{n}\sum_i \nabla_{\kappa_\mu} \min(\rho_i(\kappa_\mu)\hat{A}_i, \mathrm{clip}(\rho_i(\kappa_\mu), 1-\epsilon, 1+\epsilon)\hat{A}_i), \tag{25}$$

IV. RESULT

We implement the simulation on a desktop computer and in the simulink environment (Matlab R2017a) with the ode15s solver. The body segment properties and the MTU physiological and geometric parameters are identical to those in [19], [37]. The contact forces of the agent with the ground are managed by the physical simulator of simulink. We evaluate our proposed method on our virtual human model with

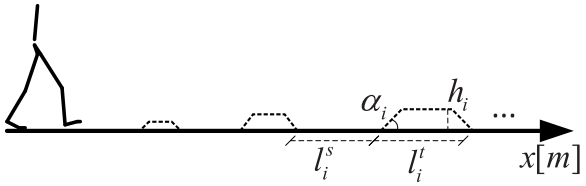
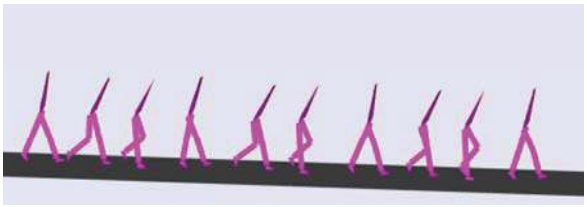


FIGURE 8. The terrain model is comprised of different sizes of trapezoidal structures with same up and down angles. The length of each structure and the space between two adjacent structures follows an even distribution.

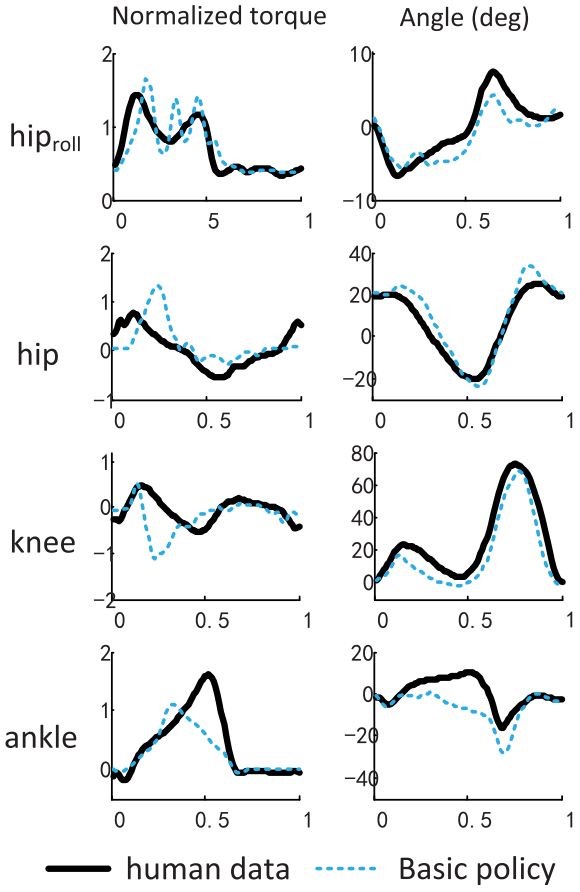
a normal walking speed 1.25 [m/s]. All of the terrains have the same height value in coronal plane and can be expressed by the 1D height-fields. Three representative classes of terrain obstacles that include steps, slopes and waves are employed in this paper. All different terrain obstacles are modeled by one or more trapezoidal structures with different height, trapezoid angle and lengths of two parallel sides. As Fig. 8 shown, the wavy ground is modeled as a series of small trapezoidal structures, and the structure length l_i^t and the space length l_i^s follow different distribution within given range. The change of the length l_i^t, the height h_i and the angle α_i for a trapezoid can generate a different obstacle. For instance, a step obstacle has large length to height ratio, and its parallel sides are set equal to each other. A slope obstacle has a longer baseline and a short topline.

In order to obtain the basic control policy parameters, we first carried out an optimization with the control model without the DNN by using the PSO algorithm. During this optimization process for basic policy, the total 76 control parameters are optimized with the constriction factor $w = 0.7$, the cognitive factor c1 $= 2.1$ and the social factor c2 $= 2.1$. The weight coefficients for reward function are set to wv $= 1$,wp $= 100$ and we $= 0.0001$ respectively. The optimal result obtained after about 3000 iterations on average. Fig. 9(a) shown the optimized walking of the NMVH with basic control policy parameters, and Fig. 9(b) given one step comparations on joint angles and torques of NMVH and real human [5].

The goal of the training for this NMVH is to learn how to walk steadily over the given terrains. When the optimization process of the basic policy is finished, the DNN in policy control layer will be trained by use of PPO algorithm for adjusting some control parameters to cope with the changes of terrain and virtual human state. The main time consumption of the learning process comes from the motion simulation of the NMVH rather than the updating of the DNN. The virtual human learns walking skills for traversing terrains with step, slope and wavy obstacles. All of the obstacles were placed randomly in the environment with spaced 0.5m to 3m apart. Fig. 10(a-c) show the learned skills of the agent walking in a single type of terrain. The steps are changed in height ranging from 0.05 to 0.2m, with lengths follow a Gaussian distribution $G(5, 0.1)$. The slope angle is evenly distributed in the range 2-10 degrees in training. The widths of obstacles in wavy terrains varied ranging from 0.3 to 1m.



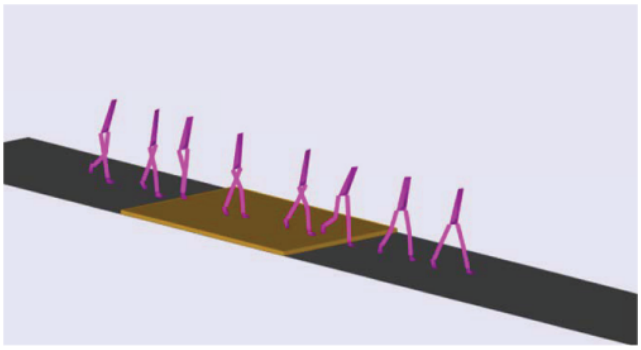
(a)



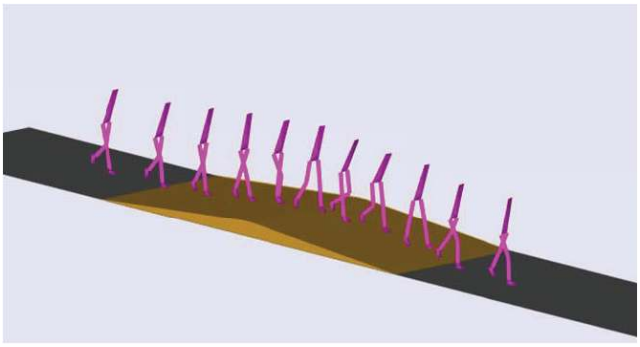
(b)

FIGURE 9. The optimized result of the basic policy for controlling NUVH to walk on flat terrain: (a) the NMVH walks on flat ground with basic control policy, (b) the trajectory comparations of joint angle and torque between NMVH and real human.

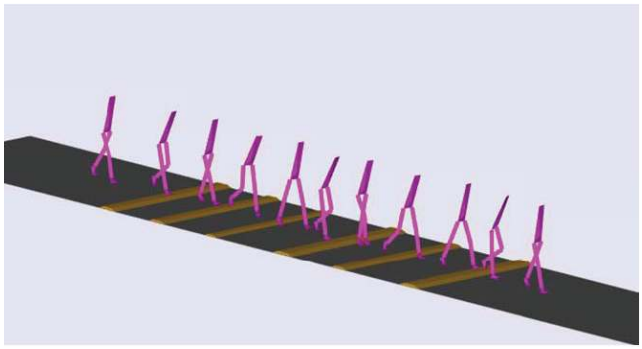
Separate policies are learned for each class of terrain. The normalized rewards of the training process for the 3 different classes of terrain are shown in Fig. 10(d). During the learning process in 3 different terrains, terrains were updated for each episode, and some tuples from basic policy were added with the main tuples coming from exploration to update the DNN in policy control layer. The experience replay memory D records the 20k most recent tuples. Updates are performed by sampling mini-batches of $n = 32$ tuples from D and applying stochastic gradient descent with momentum with a discount factor $\gamma = 0.96$. The value function learning rate α_v is set to 0.001, the policy learning rate α_μ is set to 0.001, and momentum equal to 0.9. A weight decay of 0.0005 is applied to the policy for regularization. For the action exploration,



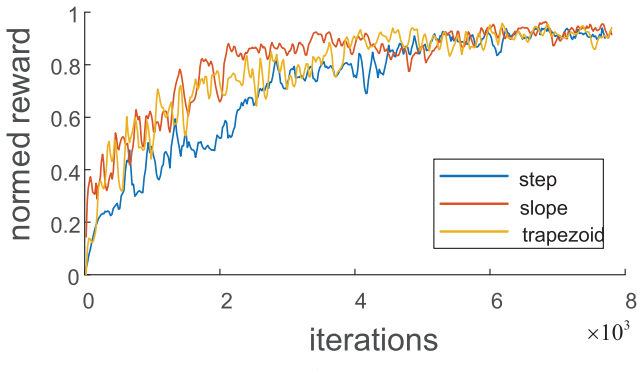
(a)



(b)



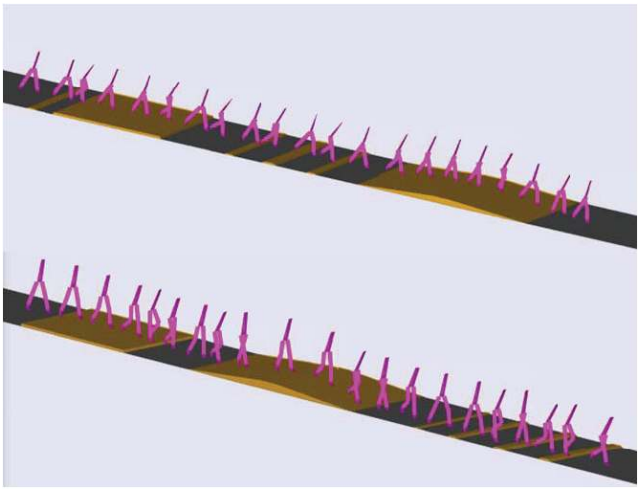
(c)



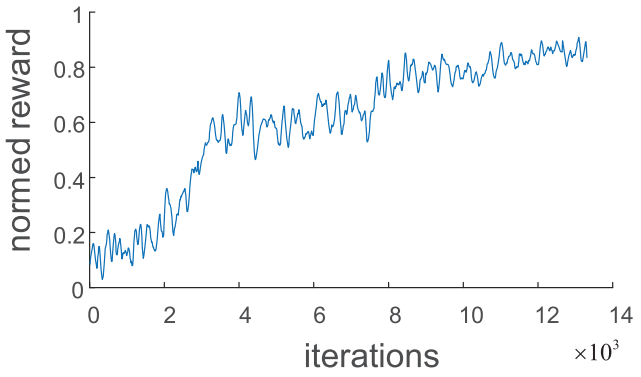
(d)

FIGURE 10. The optimized control policies of the NMVH learned on 3 classes of terrain: (a) the terrain with step obstacles only, (b) the terrain with slope obstacles only, (c) the terrain with wavy obstacles only, (d) the normalized rewards of the training process for the 3 different classes of terrain.

the covariance of the Gaussian noise is set to approximately 10% of the allowed range of values. Exploration is turned off during the evaluation of the control policies.



(a)



(b)

FIGURE 11. Results of the NMVH learned on mixed terrains: (a) the NMVH walks on the mixed terrain with the DRL policy, (b) the training process for the DRL policy.

The results show that the control policy trained by DRL in a specific class of terrain can make the NMVH able to adaptively walk on the same class of terrain. In order to achieve the capability of the NMVH to walk in complex terrains, a control policy is also learned for a mixed terrain class that randomly arranged the position and shape of the obstacles. The parameter ranges of the mixed terrains are the same as for the individual environments. Every 50 simulation steps (one episode), the learning terrain were randomly constructed and remain fixed for the course of the episode. The learned walking skills of the agent in the mixed terrain are shown in Fig. 11(a), and the normalized reward of the learning process is show in Fig. 11(b). Roughly 11000 training episodes were required to obtain good performance.

Table 1 shows the performance comparisons of our DRL trained policy and the optimized classical reflex control model (RCM) proposed by Geyer and Herr [3]. The parameters of RCM are optimized for the 4 different classes of terrain by using PSO method. The terrains which are used to test the performances of methods are constructed by successive uniform random selection of the given features with amplitudes increasing gradually. The space between two obstacles

TABLE 1. Performance(m) of two framework for characters walking on different terrain.

	performance(m)			
	step	slope	wavy	mixed
DRL policy with mixed Env.	409	471	434	435
RCM with mixed Env.	221	272	255	239
RCM with step Env.	283	128	196	165
RCM with slope Env.	94	327	153	121
RCM with wavy Env.	156	232	319	218

is randomly generated over the specified range, with the increase of the obstacles 2% every 10m distance. The mean distance before a fall is used as the performance metric, as measured across 100 epochs. Obviously, the proposed DRL policy achieved better performance than the optimized reflex control model on all 4 kinds of terrains. The RCM perform poorly when encountering unfamiliar obstacles, such as optimizing under slopes for testing in steps and optimizing under steps for testing in slopes. The DRL don't have the problem and show the same good performances for all tested environments.

V. CONCLUSION

We have presented a hierarchical biomechanical virtual human control framework for 3D bipedal walking skills. A deep neural network is integrated into the framework and trained by PPO for improving the terrain adaptive capability of the muscle-actuated 3D virtual human model. Given the results from the study in this paper, it is shown to produce robust high-level controller that can directly exploit terrain maps and work with high-dimensional state descriptions for terrain adaptive locomotion. We have also noticed that assigning fixed parameters to neural reflex network limits the motion diversity of the character. It will be important for us in the future to find ways of developing the framework of low-level controller. Moreover, we will consider improving performance of our proposed approach with some methods, such as generative adversarial imitation learning.

REFERENCES

[1] N. Thatte and H. Geyer, "Toward balance recovery with leg prostheses using neuromuscular model control," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 5, pp. 904–913, May 2016.
[2] M. A. Sharbafi, H. Barazesh, M. Iranikhah, and A. Seyfarth, "Leg force control through biarticular muscles for human walking assistance," *Frontiers NeuroRobot.*, vol. 12, p. 39, Jul. 2018.
[3] H. Geyer and H. Herr, "A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 18, no. 3, pp. 263–273, Jun. 2010.
[4] S. Song and H. Geyer, "A neural circuitry that emphasizes spinal feedback generates diverse behaviours of human locomotion," *J. Physiol.*, vol. 593, no. 16, pp. 3493–3511, Aug. 2015.
[5] J. M. Wang, S. R. Hamner, S. L. Delp, and V. Koltun, "Optimizing locomotion controllers using biologically-based actuators and objectives," *ACM Trans. Graph.*, vol. 31, no. 4, p. 25, Jul. 2012.
[6] X. B. Peng, G. Berseth, K. Yin, and M. V. D. Panne, "Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning," *ACM Trans. Graph.*, vol. 36, no. 4, p. 41, Jun. 2017.

[7] T. Geijtenbeek and N. Pronost, "Interactive character animation using simulated physics: A state-of-the-art review," *Comput. Graph. Forum*, vol. 31, pp. 2492–2515, Dec. 2012.
[8] K. Yin, K. Loken, and M. Van de Panne, "Simbicon: Simple biped locomotion control," *ACM Trans. Graph.*, vol. 26, p. 105, Jul. 2007.
[9] L. Liu, M. Van De Panne, and K. Yin, "Guided learning of control graphs for physics-based characters," *ACM Trans. Graph.*, vol. 35, no. 3, p. 29, 2016.
[10] R. Desai and H. Geyer, "Robust swing leg placement under large disturbances," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2012, pp. 265–270.
[11] T. Geijtenbeek, M. van de Panne, and A. F. van der Stappen, "Flexible muscle-based locomotion for bipedal creatures," *ACM Trans. Graph.*, vol. 32, no. 6, p. 206, 2013.
[12] Q. Nguyen, A. Agrawal, X. Da, W. C. Martin, H. Geyer, J. Grrizzle, and K. Sreenath, "Dynamic walking on randomly-varying discrete terrain with one-step preview," *Robot., Sci. Syst. XIII*, Jul. 2017. [Online]. Available: https://www.ri.cmu.edu/publications/dynamic-walking-randomly-varying-discrete-terrain-one-step-preview/
[13] A. V. Hill, "The heat of shortening and the dynamic constants of muscle," *Proc. R. Soc. Lond. B, Biol. Sci.*, vol. 126, no. 843, pp. 136–195, 1938.
[14] S. L. Delp, J. P. Loan, M. G. Hoy, F. E. Zajac, E. L. Topp, and J. M. Rosen, "An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures," *IEEE Trans. Biomed. Eng.*, vol. 37, no. 8, pp. 757–767, Aug. 1990.
[15] J. L. Hicks, M. H. Schwartz, A. S. Arnold, and S. L. Delp, "Crouched postures reduce the capacity of muscles to extend the hip and knee during the single-limb stance phase of gait," *J. Biomech.*, vol. 41, no. 5, pp. 960–967, 2008.
[16] E. Bizzi, V. C. K. Cheung, A. D'Avella, P. Saltiel, and M. C. Tresch, "Combining modules for movement," *Brain Res. Rev.*, vol. 57, pp. 125–133, Jan. 2008.
[17] S. Harkema, Y. Gerasimenko, J. Hodes, P. J. Burdick, C. Angeli, Y. Chen, C. Ferreira, A. Willhite, E. Rejc, P. R. G. Grossman, P. V. R. Edgerton, "Effect of epidural stimulation of the lumbosacral spinal cord on voluntary movement, standing, and assisted stepping after motor complete paraplegia: A case study," *Lancet*, vol. 377, pp. 1938–1947, Jun. 2011.
[18] G. N. Orlovsky, T. G. Deliagina, and S. Grillner, *Neuronal Control of Locomotion: From Mollusc to Man*. New York, NY, USA: Oxford Univ. Press, 1999.
[19] H. Geyer, A. Seyfarth, and R. Blickhan, "Positive force feedback in bouncing gaits?" *Proc. R. Soc. Lond. B, Biol. Sci.*, vol. 270, no. 1529, pp. 2173–2183, Oct. 2003.
[20] D. F. B. Haeufle, M. Günther, R. Blickhan, and S. Schmitt, "Can quick release experiments reveal the muscle structure? A bionic approach," *J. Bionic Eng.*, vol. 9, no. 2, pp. 211–223, 2012.
[21] R. Tedrake, T. W. Zhang, and H. S. Seung, "Stochastic policy gradient reinforcement learning on a simple 3D biped," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, vol. 3, Sep./Oct. 2004, pp. 2849–2854.
[22] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*. [Online]. Available: https://arxiv.org/abs/1509.02971
[23] X. B. Peng, G. Berseth, and M. Van de Panne, "Dynamic terrain traversal skills using reinforcement learning," *ACM Trans. Graph.*, vol. 34, no. 4, p. 80, 2015.
[24] J. Schulman, S. Levine, P. Moritz, M. Jordan, and P. Abbeel, "Trust region policy optimization," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1889–1897
[25] X. Guo, S. Singh, H. Lee, R. L. Lewis, and X. Wang, "Deep learning for real-time Atari game play using offline Monte-Carlo tree search planning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3338–3346.
[26] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.
[27] N. Heess, G. Wayne, D. Silver, T. Lillicrap, T. Erez, and Y. Tassa, "Learning continuous control policies by stochastic value gradients," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2944–2952.
[28] X. B. Peng, G. Berseth, and M. Van de Panne, "Terrain-adaptive locomotion skills using deep reinforcement learning," *ACM Trans. Graph.*, vol. 35, no. 4, p. 81, 2016.
[29] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Optimizing walking controllers," *ACM Trans. Graph.*, vol. 28, no. 5, p. 168, 2009.
[30] M. J. Grey, M. Ladouceur, J. B. Andersen, J. B. Nielsen, and T. Sinkjær, "Group II muscle afferents probably contribute to the medium latency soleus stretch reflex during walking in humans," *J. Physiol.*, vol. 534, no. 3, pp. 925–933, 2001.
[31] M. Knikou and W. Z. Rymer, "Effects of changes in hip joint angle on H-reflex excitability in humans," *Exp. Brain Res.*, vol. 143, no. 2, pp. 149–159, 2002.
[32] S. Meunier, A. Penicaud, E. Pierrot-Deseilligny, and A. Rossi, "Monosynaptic Ia excitation and recurrent inhibition from quadriceps to ankle flexors and extensors in man," *J. Physiol.*, vol. 423, no. 1, pp. 661–675, 1990.
[33] R. Desai and H. Geyer, "Muscle-reflex control of robust swing leg placement," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2013, pp. 2169–2174.
[34] S. Berger, J. van den Kieboom, R. Ronsse, and A. J. Ijspeert, "Energy consumption optimization and stumbling corrective response for bipedal walking gait," M.S thesis, Biorob Lab., Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2011.
[35] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*. [Online]. Available: https://arxiv.org/abs/1707.06347
[36] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA, MIT Press, 2018.
[37] S. Song and H. Geyer, "Generalization of a muscle-reflex control model to 3d walking," in *Proc. 35th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2013, pp. 7463–7466.

JIANPENG WANG is currently pursuing the Ph.D. degree with the School of Instrument Science and Engineering, Southeast University, China. His research interests include virtual character motion control and reinforcement learning.

WENHU QIN received the Ph.D. degree from Southeast University, China, in 2005, where he is currently a Professor with the vehicle safety and virtual reality lab. He has more than 30 journal papers, ten conference papers, and a book. He holds three patents. His research interests include vehicle safety, virtual reality, crowd simulation, and road traffic accident reconstruction.

LIBO SUN received the Ph.D. degree from the School of Computer Science and Technology, Tianjin University, in 2012. She was a Visiting Scholar and a Postdoctoral Researcher with the Center for Human Modeling and Simulation, University of Pennsylvania, from 2009 to 2011 and from 2015 to 2017, respectively. She is currently an Associate Professor with the vehicle safety and virtual reality lab, Southeast University, China. Her research interests include computer animation, virtual reality, and crowd simulation.