2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)
Toronto, Canada. October 15-17, 2019

A deep reinforcement learning based approach towards generating human walking behavior with a neuromuscular model

Akhil S Anand[1], Guoping Zhao[2], Hubert Roth[3] and Andre Seyfarth[4]

Abstract— **A gait model capable of generating human-like walking behavior at both the kinematic and the muscular level can be a very useful framework for developing control schemes for humanoids and wearable robots such as exoskeletons and prostheses. In this work we demonstrated the feasibility of using deep reinforcement learning based approach for neuromuscular gait modelling. A lower limb gait model consists of seven segments, fourteen degrees of freedom, and twenty two Hill-type muscles was built to capture human leg dynamics and the characteristics of muscle properties. We implemented the proximal policy optimization algorithm to learn the sensory-motor mappings (control policy) and generate human-like walking behavior for the model. Human motion capture data, muscle activation patterns and metabolic cost estimation were included in the reward function for training. The results show that the model can closely reproduce the human kinematics and ground reaction forces during walking. It is capable of generating human walking behavior in a speed range from $0.6\,\mathrm{m/s}$ to $1.2\,\mathrm{m/s}$. It is also able to withstand unexpected hip torque perturbations during walking. We further explored the advantages of using the neuromuscular based model over the ideal joint torque based model. We observed that the neuromuscular model is more sample efficient compared to the torque model.**

I. INTRODUCTION

The gait model capable of reproducing human-like locomotion can help us further understand the human locomotion control scheme which can be used for developing bipedal robots and wearable robots (e.g. exoskeletons, prostheses, etc.). For instance, with a simple inverted pendulum model, it has been shown that the human-like bipedal walking gait can be achieved passively (without active control) because of the natural dynamics of the human body [1]. It has also been found that both human walking and running gait can be described by a simple spring loaded inverted pendulum model [2], [3]. Several legged robots were developed and successfully demonstrated the control benefits of these models [1], [4], [5]. Besides, the bio-inspired conceptual model also shows benefits for the exoskeleton control [6]. However, although these simplified template models can generate human-like gait (in terms of e.g. the centre of mass movement, step length/frequency, etc.), their capability of reproducing human-like rich locomotion behaviors (e.g. stair/slope climbing, acceleration/deceleration, etc.) is very limited.

Recently, Song and Geyer [7] demonstrated that the diverse behaviours of human locomotion can be generated with a complex neuromuscular gait model using a neural circuitry which emphasizes the muscle reflexes. They also demonstrated that the model can produce human-like immediate changes in the muscle activation of some muscle groups [8]. However, the architecture of the neural circuitry used in the model was hand-crafted. The model performance could be further improved if we explore the circuitry (connections of reflex pathways) with a systematic approach. In [9], authors developed a lower-limb musculoskeletal model based on contact invariant optimization primarily for animating human activities driven by lower body.

[10]–[12] presented different approaches to muscle based locomotion controllers, while [13], [14] presented a similar approach as our work using deep reinforcement learning by learning muscle activation pattern and joint torque pattern respectively. Peng *et al.* [15], [16] and Merel *et al.* [10] showed that deep reinforcement learning (deep-RL) is very useful approach in developing robust controllers for complex locomotive systems. They also demonstrated the capability of deep-RL in learning a broad range of challenging locomotion skills using kinematic data. Another closely matching work from Peng *et al.* [17], demonstrated learning 2D muscle actuated bipedal locomotion using deep-RL. They identified that the local feedback provided by high-level action parameterizations can significantly impact the learning, robustness, and motion quality of the resulting policies. Our work is focused on learning the individual specific walking behaviour at various walking speeds by directly learning the muscle activation pattern.

RL had its major successes in the discrete domain problems such as computer games [18], but human locomotion needs to be solved as a continuous control problem. RL is of advantage in solving continuous domain problems ever since the latest developments in policy gradient based methods [19]–[22]. Policy gradients were a breakthrough in the continuous domain, but still limited by many factors such as the learning rate, sample efficiency etc. Many approaches tried to eliminate these flaws which resulted in the development of algorithms such as TRPO [20], ACER [23], PPO [19] etc. All these methods have their own trade-offs, ACER is by far more complicated than PPO, requiring the addition of code for off-policy corrections and a replay buffer with very

[1] Akhil S Anand is with dept. of engineering cybernetics at Norges teknisk-naturvitenskaplige universitet - NTNU, 7034, Trondheim, Norway. akhil.s.anand@ntnu.no

[2] Guoping Zhao is with Lauflabor Locomotion Laboratory at Technische Universität Darmstadt, 64289, Darmstadt, Germany. zhao@sport.tu-darmstadt.de

[3] Hubert Roth is with the Institute of Control Engineering at Universität Siegen, 57076, Siegen, Germany. hubert.roth@uni-siegen.de

[4] Andr Seyfarth is with Lauflabor Locomotion Laboratory at Technische Universität Darmstadt, 64289, Darmstadt, Germany. seyfarth@sport.tu-darmstadt.de

978-1-5386-7630-1/19/$31.00 ©2019 IEEE

marginal advantage in the results tested on Atari benchmark by Open-AI [24]. Considering the above mentioned trade-offs, we decided to use the Proximal Policy Optimization (PPO) [19] algorithm to address our problem regarding human locomotion.

The aim of this paper is to investigate the feasibility of using deep-RL for generating human walking with a neuromuscular gait model. Here, we present a deep-RL based approach towards the development of a human gait model capable of producing individual-specific 3D walking gait at the kinematic, the kinetic and the muscular levels. Although the major focus of our work is on developing deep-RL based human walking gait, we also explore the advantage of learning a muscle-based control over torque-based control in terms of sample efficiency.

The approach followed in this paper is, (i) conducting human experiments to collect the individual kinematic and kinetic data and providing a dataset for deep-RL, (ii) setting up a musculoskeletal gait model to perform deep-RL, (iii) conducting deep-RL to generate human-like kinematics and to optimize energetics, (iv) testing model against robustness, and (v) comparing the sample efficiency of muscle-based and torque-based control.

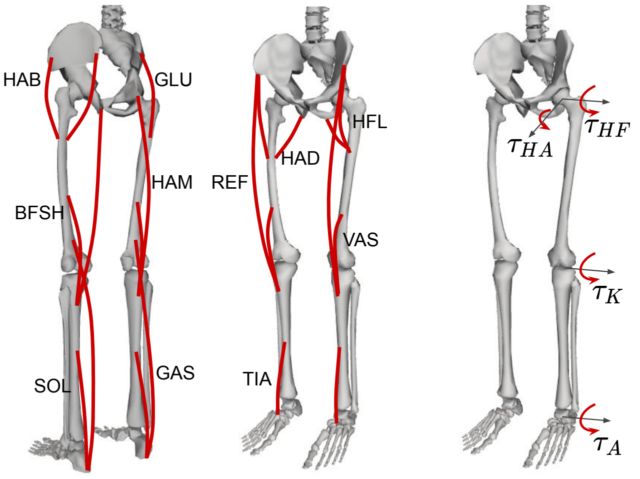
II. Methods

A. Human experiments

Human treadmill walking experiments were conducted with one subject (male, 27 years, height $1.75\,\mathrm{m}$, weight $66\,\mathrm{kg}$) to acquire the data of lower-limb joint kinematics and ground reaction forces (GRFs) in a walking speed range from $0.6\,\mathrm{m/s}$ to $1.2\,\mathrm{m/s}$. The subject provided their informed consent for the experiment. The study design and protocol were approved by the ethical committee of TU Darmstadt. The experimental data was processed to prepare the dataset for reinforcement learning. In total, the dataset contains 1200 walking steps each sampled at a frequency of $200\,\mathrm{Hz}$. One step in the dataset contains all the individual lower body joint kinematics from one foot touch-down to the contra-lateral foot touch-down.

B. Modelling

The musculoskeletal model used in this study is a lower limb human model with seven segments and twenty two muscles. The model was adopted from [7] and implemented in MuJoCo [25] to achieve high simulation speed. Then the model was integrated with Open-AI Gym to facilitate easy implementation of deep-RL. The model is $1.8\,\mathrm{m}$ tall, has a weight of $66\,\mathrm{kg}$ and fourteen degrees of freedom (six global DOFs for the trunk and eight internal joint DOFs). The model is $5\,\mathrm{cm}$ taller than the subject, but we believe the effect of such a small height difference on the joint kinematics is negligible. The physical properties, muscle-tendon-units (MTU) and the muscle properties are similar to the model from [7] except for the foot. The foot is modelled as a cuboid (width $10\,\mathrm{cm}$, length $25\,\mathrm{cm}$ and height $6\,\mathrm{cm}$) with four ground contact points. The eleven muscle groups of each leg are shown in Fig. 1(a). The torque controlled model also posses same weight, segment dimensions and degrees of freedoms. The model is actuated with eight ideal torque actuators (four for each leg) as depicted in Fig. 1(b).



(a) neuromusculoskeletal gait model

(b) torque gait model

Fig. 1. Schematics of the musculoskeletal model and the torque model. (a) Lower-limb musculoskeletal model with eleven muscle groups per leg. The muscle groups are hip abductors (HAB), hip adductors (HAD), hip flexors (HFL), glutei (GLU), hamstrings (HAM), rectus femoris (REF), vastii (VAS), biceps femoris short head (BFSH), gastrocnemius (GAS), soleus (SOL), and tibialis (TIA). The HAM, REF and GAS are biarticular muscles. The HAB, HAD, HFL, GLU, VAS, BFSH and TIA are monoarticular muscles. (b) Lower-limb torque controlled bipedal model with 8 torque actuators. There are 4 joint torque actuators for each leg which are (i) hip flexion/extension τ_{HF}, (ii) hip adduction/abductionτ_{HA}, (iii) knee flexion/extension τ_K and (iv) ankle dorsiflexion/plantar flexion τ_A. The torque values could be both positive and negative.

C. Deep-RL implementation

In the final implementation with PPO algorithm, separate network architectures and hyper-parameters are chosen for both muscle-based and torque-based model as they differ in the state-action space dimension and characteristics. The input state space for the torque model contains the joint positions θ, joint angular velocities $\dot{\theta}$, GRFs grf and target walking velocity v. The muscle-based model has additional input state, which are muscle force f, muscle length l, muscle velocity v_m and muscle activations a. The input to the muscle model is the muscle stimulation u. In the case of the torque model the inputs are the joint torques τ to the 8 joints.

Both the policy and value function architecture are defined using neural networks shown in Fig. 2. All the hyperparameters for the learning process are shown in the Table I. The algorithm is implemented with 40 threads (workers), collecting data by acting on the environment and training the neural network on a single GPU. The pseudo-code of the implementation is shown in Algorithm 1. The input state vector and the scalar reward values are normalized using its running mean. And The standard deviation of the states are clipped to the range [-10, 10].

In Algorithm 1, θ and ϕ are the policy and value function (baseline) parameters. N is the total number of time-steps,

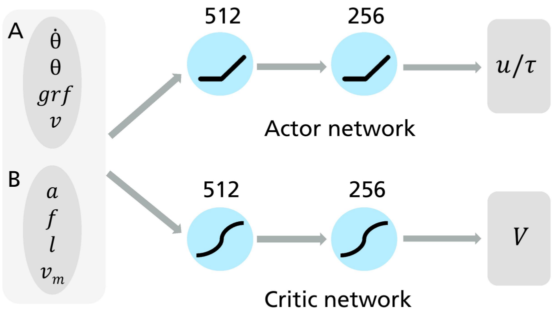


Fig. 2. The network consists of two fully connected hidden layers of size 512 and 256 respectively. The actor (policy) and the critic have the same network structure. The activation function for actor and critic networks are ReLU and tanh respectively for both muscle and torque models. The input set A is the input for torque model and A,B together is the input for the muscle model. u and τ denotes the muscle activations and joint torques for muscle and torque models respectively. V denotes the state value from the critic

TABLE I

HYPERPARAMETERS USED FOR MUSCLE AND TORQUE MODELS

Hyperparameter	Muscle model	Torque model
No. of actors N	40	40
Samples per actor/episode n	128	128
No. of minibatches	32	32
No. of epochs	7	7
Clip factor β	0.2	0.2
GAE Parameter λ	0.95	0.95
Discount factor γ	0.99	0.99
Value function coefficient c_1	0.5	0.5
Entropy coefficient c_2	0 0.005	0 0.005
Learning rate lr	3e-5 1e-7	5e-4 1e-6

Algorithm 1 Pseudo-code for PPO implementation

```
$\theta \leftarrow$ random weights
$\phi \leftarrow$ random weights
for $n \in \{1, \dots, N\}$ do
    $\pi_{\mathrm{w}} \leftarrow \pi_\theta$
    Run W workers in parallel
    $\pi_{\mathrm{old}} \leftarrow \pi_\theta$
    for $i \in \{1, \dots, I\}$ do
        $J_{PPO}(\theta) = \sum_{t=1}^{T} \frac{\pi_\theta(a_t|s_t)}{\pi_{old}(a_t|s_t)} \hat{A}_t - \lambda KL\left[\pi_{old}|\pi_\theta\right]$
        Update $\theta$ by stochastic gradient method w.r.t. $J_{PPO}(\theta)$
    end for
    for $j \in \{1, \dots, J\}$ do
        $L_V(\phi) = -\sum_{t=1}^{T} \left(\sum_{t'>t} \gamma^{t'-t} r_{t'} - V_\phi(s_t)\right)^2$
        Update $\phi$ by a gradient method w.r.t. $L_V(\phi)$
    end for
    if $\mathrm{KL}\left[\pi_{old}|\pi_\theta\right] > \beta_{high}\mathrm{KL}_{target}$ then
        $\lambda \leftarrow \alpha\lambda$
    else if $\mathrm{KL}\left[\pi_{old}|\pi_\theta\right] < \beta_{low}\mathrm{KL}_{target}$ then
        $\lambda \leftarrow \lambda/\alpha$
    end if
end for
```

Worker

```
for $t \in \{1, \dots, T\}$ do
    Run policy $\pi_\theta$ , collecting $\{s_t, a_t, r_t\}$
    Estimate return $R_t = \sum_{t'=t} \gamma^{t'-t} r(s_{t'}, a_{t'})$
    Estimate advantages $\hat{A}_t = R_t - V_\phi(s_t)$
    Store the trajectory information
end for
```

I and J are the number of sub-iterations with policy and baseline (value function here) updates over a batch of data points. T denotes the number of data points collected per worker. λ and α are the KL regularization coefficient and the scaling term, respectively.

D. Muscle dynamics and perturbation protocols

In order to generate human-like muscle activation patterns, the model is trained with the guidance of the human experimental data from [26]. The guidance is implemented by clipping the stimulations at appropriate phases in the gait cycle. For example, in the second half of stance phase HAD and HAM muscles are completely inactive. This evidence is implemented by clipping the HAD and HAM activations to 0 during this phase. The muscle control frequency is $5\,\mathrm{kHz}$ and all the sensory feedback signals (kinematics, muscle dynamics and ground reaction forces) and the input stimulation signals for the muscle model are delayed by $15\,\mathrm{ms}$ to mimic the human sensory feedback delay [7].

The model is trained with joint torque perturbations on hip flexion/extension movements. In the training phase, random joint torque perturbations are applied on the model continuously for $50\,\mathrm{ms}$ (starting at a randomly chosen time step in the gait cycle) with a magnitude in the range from $-5\,\mathrm{N\,m}$ to $5\,\mathrm{N\,m}$. A maximum of only one perturbation is applied per gait cycle in any one of the hip joints. The probability of applying perturbation in any gait cycle is 50%. For testing the robustness, much larger joint torque perturbations in the range of [-200, 200] Nm are applied on the hip continuously for for $50\,\mathrm{ms}$. The joint torque perturbations are chosen to emulate the situation of using exosuits for assisting/perturbing walking.

Random State Initialisation (RSI) and Early Termination (ET) are very useful methods for reinforcement learning [15]. We divided each trajectory in the training dataset into 10 equal time intervals. RSI is implemented by randomly selecting trajectories during training from the reference kinematic dataset and defining the initial condition by randomly choosing from 10 equal intervals of the trajectory. The idea of ET is implemented by terminating the episode if the kinematic error exceeds a given limit. More specifically, the ET is terminated if the pelvis vertical position is lower than $0.8\,\mathrm{m}$ or higher than $1.4\,\mathrm{m}$, which corresponds to the undesired falling and jumping motion respectively.

E. Reward shaping

The reward function for the muscle and torque model contain terms which encourage imitating the kinematic trajectory, continuous stable walking, attaining a target velocity.

An additional metabolic cost reward term is included for the muscle model, but the torque model is learned without using any torque minimization term.

$$r = w_l r_l + w_k r_k + w_m r_m + w_v r_v \tag{1}$$

where r is the reward, r_l is the life bonus, r_k is the kinematic behavior bonus, r_m is the metabolic bonus and r_v is the target velocity bonus. w_l=1, w_k=4, w_m=4, w_v=1 are the weights of r_l, r_k, r_m and r_v, respectively. All these individual bonus is between 0 and 1. The total reward, r is in the range from 0 to 10. The life bonous r_l denotes the reward for walking without falling. The falling condition occur when the pelvis vertical position is out of the range [0.8, 1.4] m. The r_k term defines the reward for imitating the desired trajectory. Individual position and velocity errors between the model and the experimental data are calculated for each sampling step. These errors are:

- Foot position vector error e_{fp} which denotes the squared difference between the foot position vector of the model and the reference human trajectory data.

$$e_{fp} = [c(s_{fp}(t) - \bar{s}_{fp}(t))]^2 \tag{2}$$

Here, $s_{fp}(t)$ and $\bar{s}_{fp}(t)$ are the foot position vector of the model and the reference data respectively at time t and the scaling coefficient, c=30.

- Pelvis COM position error e_{pp} which denotes the squared difference between the pelvis COM position vector of the model and the reference data.

$$e_{pp} = [c(s_{pp}(t) - \bar{s}_{pp}(t))]^2 \tag{3}$$

Here, $s_{pp}(t)$ and $\bar{s}_{pp}(t)$ are the pelvis COM position vector of the model and the reference data respectively at time t and c=20.

- Pelvis COM velocity error e_{pv} which denotes the squared difference between the pelvis COM velocity vector of the model and the reference data.

$$e_{pv} = c[s_{pv}(t) - \bar{s}_{pv}(t)]^2 \tag{4}$$

Here, $s_{pv}(t)$ and $\bar{s}_{pv}(t)$ are the pelvis COM velocity vector of the model and the reference data respectively at time t and c=2.

- Joint angular position error e_{ap} which denotes the squared difference between all the joint angles of the model and the reference data.

$$e_{ap} = [c(\theta_{ap}(t) - \bar{\theta}_{ap}(t))]^2 \tag{5}$$

Here, $\theta_{ap}(t)$ and $\bar{\theta}_{ap}(t)$ are the array of all the joint angles of the model and the reference data respectively at time t and c=12.

- Joint angular velocity error e_{av} which denotes the squared difference between all the joint angular velocities of the model and the reference data.

$$e_{av} = [c(\theta_{av}(t) - \bar{\theta}_{av}(t))]^2 \tag{6}$$

Here, $\theta_{av}(t)$ and $\bar{\theta}_{av}(t)$ are the array of all the joint angular velocities of the model and the reference data respectively at time t and c=0.1.

All these individual errors are concatenated to form a single error vector, E as follows:

$$E = [e_{fp}, e_{pp}, e_{pv}, e_{ap}, e_{av}] \tag{7}$$

E is converted to its negative exponential and the resulting terms are summed up to get a scalar value T:

$$T = \sum \mathrm{e}^{-E} \tag{8}$$

The r_k term denotes how large is the T value compared to the limiting value of 28. It is computed as follows

$$r_k = \frac{T - T_{\text{limit}}}{T_{max} - T_{limit}} \quad \text{where } T_{limit} = 28 \ , \ \ T_{\max} = 35 \tag{9}$$

The value of r_k is between 0 to 1, where 1 denotes an exact imitation of the joint trajectory and 0 corresponds a maximum allowed deviation defined by T_{limit}. The T_{limit} is also used as the Early Termination (ET) criterion. In other words, the ET will be triggered if $T < T_{limit}$.

The metabolic rate p for the musculoskeletal model is estimated based on the muscle states according to Alexanders work [27]. The metabolic energy over a sampling step is converted to a value between 0 to 1 by taking the negative exponential with a coefficient of 1/30. The value of 1/30 is chosen by monitoring the range of p during training. The calculation of r_m as follows:

$$r_m = e^{-p/30} \tag{10}$$

The r_v term is a function of the difference between the running mean of the experimental walking speed $\bar{v}_p$ and the running mean of the model walking speed v_p.

$$r_v = \frac{\sum(e^{e_v})}{3} \quad \text{where } e_v = c[\bar{v}_p - v_p]^2 \tag{11}$$

The coefficient $c = 50$.

The implemented algorithm can be found at here [1].

III. RESULTS

A. Reproducing human gait

The learned musculoskeletal gait model is capable of generating human-like joint kinematics, muscle activation and GRF in a speed range from $0.6\,\mathrm{m/s}$ to $1.2\,\mathrm{m/s}$. It could maintain a predefined target walking speed with an error bound of $0.1\,\mathrm{m/s}$ when it is initialized with the desired speed. The kinematic behavior generated by the muscle model at $1.2\,\mathrm{m/s}$ walking speed in comparison to the experimental data is shown in the Fig. 3. The comparison shows a close correlation (correlation values R between 0.82 to 0.98) between the model and the experimental data for all the lower limb joint angles. Compared to the joint angles, the joint angular velocity patterns of the model are less similar to the experimental data (R values are between 0.46 and 0.92). The model is able to reproduce the asymmetric characteristics in the human data (e.g. hip frontal joint angle). The kinematic results can be visualized in the video [1].

[1] http://lauflabor.ifs-tud.de/doku.php?id=projects:projects_learnwalk

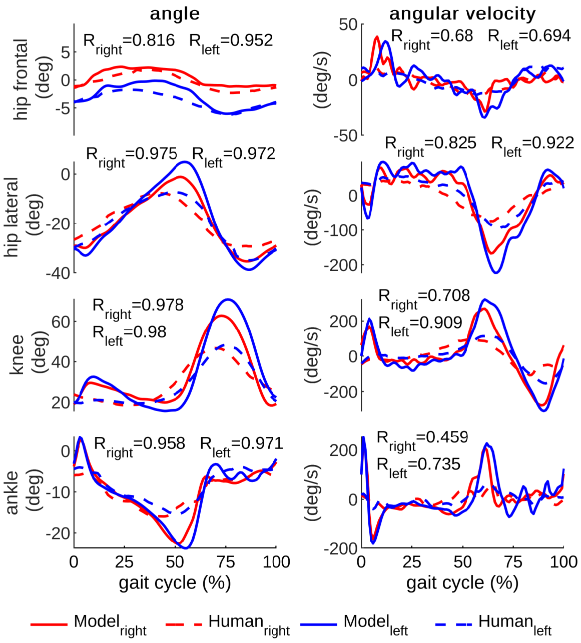


Fig. 3. Joint angle and angular velocity of the musculoskeletal gait model (in solid lines) and the human experimental data (in dashed lines) at the walking speed of $1.2\,\mathrm{m/s}$. The data are the mean of 100 steps of steady state walking during a gait cycle (touch-down to touch-down). The red and blue color denote the right and left leg joint data, respectively. The R_{right} and R_{left} denote the cross correlation values (R) for right and left leg joints respectively.

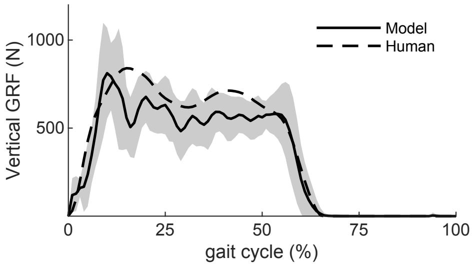


Fig. 4. Vertical GRF of the musculoskeletal gait model (the solid line) and the human experimental data (the dashed line) during walking. The mean (over 100 walking steps) GRF for a walking speed of $1.2\,\mathrm{m/s}$ is shown for the musculoskeletal gait model. The error band denotes ± 1 standard deviation.

The GRFs generated by the gait model are not as smooth as in the human experimental data. The mean (over 100 steps of walking) of the vertical GRF from the gait model (muscle based model) has a high correlation with the experimental data with a R value of 0.98 at a walking speed of $1.2\,\mathrm{m/s}$ (shown in Fig. 4). The gait model is trained to generate human-like muscle activations through stimulation clipping and optimizing for minimum metabolic cost. The muscle activation patterns generated by the gait model are shown in Fig. 5. The overall muscle activation patterns are similar to the experimental data found in the literature [26].

The musculoskeletal gait model produces robust walking. It can recover from the perturbation up to $200\,\mathrm{N\,m}$ on the hip. The perturbation response exhibited by the gait model is highly dependent on the timing of the perturbation. For touch-down and take-off conditions the perturbation response is random as it is not exhibiting the expected behaviour of model becoming increasingly unstable with increasing perturbation magnitude. For instance, the performance drops when high extension torques are applied to the swing leg.

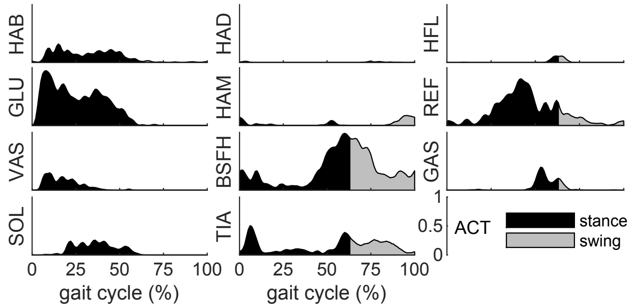


Fig. 5. The muscle mean activation of the musculoskeletal gait model at a walking speed of $1.2\,\mathrm{m/s}$. The stance phase and the swing phase are denoted as black and grey area.

B. *Muscle control vs torque control*

After training, the torque based gait model is also able to imitate the human joint kinematics (sample result is shown in Fig. 6). Compared with the torque based model, the muscle based model shows slightly better performance in reproducing human kinematic data. For example the mean (of left and right) R values of the muscle based model are higher than the torque based model for hip adduction/abduction, hip flexion, knee flexion and ankle angle at a walking speed of $1.2\,\mathrm{m/s}$.

The learning progress is depicted in the learning curves in Fig. 7. The muscle model achieves a mean return of 1688 after 10 million time-steps. But the torque model could achieve only a mean return of 1497 after 27.65 million time-steps. Although the same mean return values for both models do not correspond to exactly same behaviour, the returns are comparable as the single step rewards for both the models are scaled between 0 to 10.

IV. DISCUSSIONS

We derived a sensory motor mapping of human walking behavior at the spinal cord level using an artificial neural network. The learned muscle and torque model is able to closely follow the joint angles from experimental data. When comparing the results of our model at $1.2\,\mathrm{m/s}$ with the model from Song and Geyer [7], our model has a R values of 0.832 and 0.946 for the left and right leg, respectively, compared to 0.54 in their model on reproducing the hip adduction/abduction movement. Also for ankle (frontal plane) movements, our model has a R value of 0.96 compared to 0.46 in their model at $1.2\,\mathrm{m/s}$. Our model could also learn the left-right asymmetries of the human subject very closely, which is evident in the hip movement on the frontal plane. These are considerable improvements over the existing gait models.

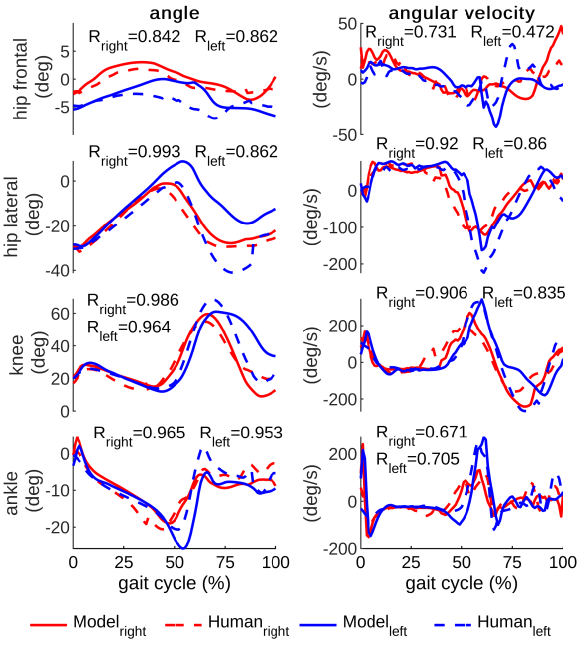


Fig. 6. Joint angle and angular velocity of the torque-based gait model (in solid lines) and the human experimental data (in dashed lines) at the walking speed of 1.2 m/s. The red and blue color denote the right and left leg joint data, respectively. R denotes the cross-correlation value.

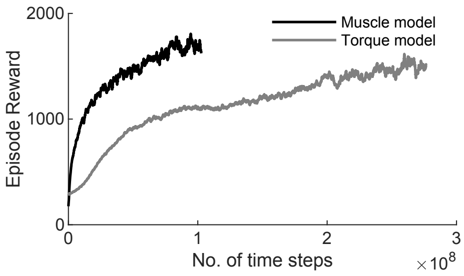


Fig. 7. Learning curve of the muscle model (in back) and the torque model (in grey) training.

Humans tend to walk with a preferred step frequency at a given speed to minimize the metabolic cost [28]. Thus by optimizing the model for minimum metabolic cost, it enables the model to mimic human-like energetics. The muscle patterns were optimized using metabolic cost minimization and also by clipping the stimulation inputs to the model based general human muscle activation reference data [7]. The HAB, HAD, HFL, GLU, VAS, SOL and TIA muscles have a good correlation to human muscle activation pattern. In contrast, the activation patterns in HAM, REF, BFSH and GAS muscle groups clearly deviate from human patterns. The low activations in GAS is compensated by the high activations of BFSH as these muscles together contribute to knee flexion. Similarly, the higher activation levels of REF muscle group in our model could correspond to the very low activations of HAM muscle group as both contribute to hip extension. The muscle dynamics are different from human data as the metabolic cost did not attain global minimum. This would demand further optimization.

The GRF profile is an important characteristic in human locomotion [29]–[31]. This is visible in the learning procedure as the GRF feedback has a large influence on the final policy. The GRF profile generated by the model is not as smooth as the human experimental data. This could be due to the rigid ground contact model in the simulation. It could be resolved by using a more realistic ground contact model. The three-segmented foot model used in OpenSim models could be adopted providing foot the flexibility of a smooth touch-down, roll over and push-off compared to the single rigid foot element with four contact points as in this work.

Compared to torque based model, the muscle based model learned a policy which has higher sample efficiency during training. This result is similar to the findings in other studies, showing the advantages of muscle based control for locomotion tasks over torque based control [15], [32]–[35]. No detailed comparison between the performances of torque based and muscle based control for bipedal locomotion is carried out in this study. But further aspects of muscle based and torque based control will be addressed in the future studies.

The perturbation response study for testing the model's robustness generated unexpected results. The response behaviour doesn't show any correspondence with the model stability and the perturbation torques, rather it is highly random. For example model become unstable after 20 steps when a perturbation torque of 40Nm is applied, but model is able to walk up-to 100 steps after applying perturbation torque of 150Nm. The reason for this random perturbation response is not yet clear and needs to be addressed in the future research. This could be a result of the nonlinear policy learned and the passive dynamics of the muscles, which is making the model capable of stabilizing even at very high perturbation toques at the hip such as ±200 Nm. The muscle model could withstand such high perturbations although it is trained with very low perturbation torques of ±5 Nm. This is because the muscle based control is taking advantage of its passive dynamics of the musculoskeletal structure to learn a robust policy. But model could not learn the acceleration/deceleration behaviour of the human subject while walking. This is because of the lack of acceleration/deceleration behaviour in the training data. This could be improved by using training data acquired on an well designed experiment where the subject is asked to accelerate and decelerate between various speeds while walking.

OpenSim models were not used in this study because of the computational complexity. Such models could be useful for a future study with larger computational resources. In this study we use a general purpose neuromuscular model (e.g. muscle parameters, segment length and mass distributions were obtained from literature) because it is difficult to obtain subject specific parameters. The model can better reproduce individualized gait if the model parameters are individualized. The torque model was not optimized for energy efficiency because including the energy efficiency term would not lead to higher sample efficiency. A detailed analysis of torque based vs muscle based control could be addressed in the future study.

V. CONCLUSIONS

In this study we explored the idea of learning a gait model to perform human-like walking using deep-RL. For learning the gait model, a reward function was designed based on the kinematics, target walking speed, stability, and metabolic cost. The learned gait model is capable of reproducing human walking kinematics and kinetics robustly at a defined target velocity. The results demonstrate the advantages of modelling the human gait using deep-RL. The results also show that the muscle-based control is superior to the torque-based control in terms of learning sample efficiency. Our future goal is to develop a deep-RL based individualized walking gait model that is more robust and capable of reproducing human response to unexpected perturbations. And with the learned model we aim to identify optimal control schemes for human assistive devices such as lower limb exoskeletons and prostheses.

ACKNOWLEDGMENT

We thank Dr. Martin Grimmer (TU Darmstadt) for helping us in revising the manuscript and conducting the human experiments. This work was partially supported by the DFG-Funded EPA Project under Grant No. AH307/2-1 and Grant No. SE1042/29-1.

REFERENCES

[1] T. McGeer, "Passive dynamic walking," *The International Journal of Robotics Research*, vol. 9, no. 2, pp. 62–82, 1990.

[2] R. Blickhan, "The spring-mass model for running and hopping," *Journal of Biomechanics*, vol. 22, no. 1112, pp. 1217 – 1227, 1989.

[3] H. Geyer, A. Seyfarth, and R. Blickhan, "Compliant leg behaviour explains basic dynamics of walking and running." *Proceedings. Biological sciences / The Royal Society*, vol. 273, no. 1603, pp. 2861–7, Nov. 2006.

[4] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science*, vol. 307, no. 5712, pp. 1082–1085, 2005.

[5] C. Hubicki, J. Grimes, M. Jones, D. Renjewski, A. Sprwitz, A. Abate, and J. Hurst, "Atrias: Design and validation of a tether-free 3d-capable spring-mass bipedal robot," *The International Journal of Robotics Research*, vol. 35, no. 12, pp. 1497–1521, 2016.

[6] G. Zhao, M. Sharbafi, M. Vlutters, E. van Asseldonk, and A. Seyfarth, "Template model inspired leg force feedback based control can assist human walking," in *2017 International Conference on Rehabilitation Robotics (ICORR)*, July 2017, pp. 473–478.

[7] S. Song and H. Geyer, "A neural circuitry that emphasizes spinal feedback generates diverse behaviours of human locomotion," *The Journal of physiology*, vol. 593, no. 16, pp. 3493–3511, 2015.

[8] ——, "Evaluation of a neuromechanical walking control model using disturbance experiments," *Frontiers in Computational Neuroscience*, vol. 11, p. 15, 2017.

[9] I. Mordatch, J. M. Wang, E. Todorov, and V. Koltun, "Animating human lower limbs using contact-invariant optimization," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, p. 203, 2013.

[10] J. Merel, A. Ahuja, V. Pham, S. Tunyasuvunakool, S. Liu, D. Tirumala, N. Heess, and G. Wayne, "Hierarchical visuomotor control of humanoids," *arXiv preprint arXiv:1811.09656*, 2018.

[11] J. M. Wang, S. R. Hamner, S. L. Delp, and V. Koltun, "Optimizing locomotion controllers using biologically-based actuators and objectives," *ACM transactions on graphics*, vol. 31, no. 4, 2012.

[12] T. Geijtenbeek, M. Van De Panne, and A. F. Van Der Stappen, "Flexible muscle-based locomotion for bipedal creatures," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, p. 206, 2013.

[13] Y. Lee, M. S. Park, T. Kwon, and J. Lee, "Locomotion control for many-muscle humanoids," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 6, p. 218, 2014.

[14] Y. Jiang, T. Van Wouwe, F. De Groote, and C. K. Liu, "Synthesis of biologically realistic human motion using joint torque actuation," *arXiv preprint arXiv:1904.13041*, 2019.

[15] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *arXiv preprint arXiv:1804.02717*, 2018.

[16] X. B. Peng, G. Berseth, K. Yin, and M. Van De Panne, "Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 41, 2017.

[17] X. B. Peng and M. van de Panne, "Learning locomotion skills using deeprl: Does the choice of action space matter?" in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 2017, p. 12.

[18] H. S. Chang, M. C. Fu, J. Hu, and S. I. Marcus, "Google deep minds alphago," *OR/MS Today*, vol. 43, no. 5, pp. 24–29, 2016.

[19] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[20] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International Conference on Machine Learning*, 2015, pp. 1889–1897.

[21] 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," *arXiv preprint arXiv:1509.02971*, 2015.

[22] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, 2016, pp. 1928–1937.

[23] Z. Wang, V. Bapst, N. Heess, V. Mnih, R. Munos, K. Kavukcuoglu, and N. de Freitas, "Sample efficient actor-critic with experience replay," *arXiv preprint arXiv:1611.01224*, 2016.

[24] P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, Y. Wu, and P. Zhokhov, "Openai baselines," https://github.com/openai/baselines, 2017.

[25] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 5026–5033.

[26] J. Perry, J. Burnfield, and J. Burnfield, *Gait Analysis: Normal and Pathological Function*. SLACK, 2010.

[27] R. M. Alexander, "Optimum muscle design for oscillatory movements," *Journal of theoretical Biology*, vol. 184, no. 3, pp. 253–259, 1997.

[28] M. Y. Zarrugh, F. N. Todd, and H. J. Ralston, "Optimization of energy expenditure during level walking," *European Journal of Applied Physiology and Occupational Physiology*, vol. 33, no. 4, pp. 293–306, Dec 1974. [Online]. Available: https://doi.org/10.1007/BF00430237

[29] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, "The development of honda humanoid robot," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 2. IEEE, 1998, pp. 1321–1326.

[30] J. P. Hunter, R. N. Marshall, and P. J. McNair, "Relationships between ground reaction force impulse and kinematics of sprint-running acceleration," *Journal of applied biomechanics*, vol. 21, no. 1, pp. 31–43, 2005.

[31] T. S. Keller, A. Weisberger, J. Ray, S. Hasan, R. Shiavi, and D. Spengler, "Relationship between vertical ground reaction force and speed during walking, slow jogging, and running," *Clinical biomechanics*, vol. 11, no. 5, pp. 253–259, 1996.

[32] A. Cruz Ruiz, C. Pontonnier, N. Pronost, and G. Dumont, "Muscle-based control for character animation," in *Computer Graphics Forum*, vol. 36, no. 6. Wiley Online Library, 2017, pp. 122–147.

[33] J. Schröder, K. Kawamura, T. Gockel, and R. Dillmann, "Improved control of a humanoid arm driven by pneumatic actuators," *Proceedings of Humanoids 2003*, 2003.

[34] T. Komura, Y. Shinagawa, and T. L. Kunii, "A muscle-based feedforward controller of the human body," in *Computer Graphics Forum*, vol. 16, no. 3. Wiley Online Library, 1997, pp. C165–C176.

[35] X. Shen, "Nonlinear model-based control of pneumatic artificial muscle servo systems," *Control Engineering Practice*, vol. 18, no. 3, pp. 311–317, 2010.