Learning to Walk: Phasic Policy Gradient for Healthy and Impaired Musculoskeletal Models

Efstratios Mytaros, Vishal Raveendranathan, and Raffaella Carloni

***Abstract*— This paper focuses on deep reinforcement learning for physics-based musculoskeletal simulations of healthy and impaired (transfemoral amputee) models during normal level ground walking. The proposed method builds upon the phasic policy gradient framework (which expands the proximal policy optimization method by separating policy and value function training) and combines it with imitation learning. The optimization of the deep neural network is carried out on the healthy model and, then, applied to the transfemoral amputee model. The healthy model learns to develop gait patterns and muscle forces similar to that from a public experimental dataset of healthy subjects, while the transfemoral model shows a plausible gait pattern in accordance with the introduced muscle impairment. The results also show that the phasic policy gradient optimization significantly improves the simulation of the two models during level-ground walking when compared to proximal policy optimization.**

I. Introduction

Computer simulations of physics-based models provide a valuable aid in the research on human and robotic locomotion. In this broad field, deep reinforcement learning (DRL) has recently proved to have high potential in teaching physics-based models to efficiently perform locomotion tasks. In [1], a hexapod robot can learn to walk across different terrains in the MuJoCo environment. In [2] a NAO humanoid robot can generate different gait patters by means of Q-learning. In [3] and [4], several robotic models with varying limbs and joints are tested in MuJoCo with different DRL algorithms to learn locomotion patterns. More complex algorithms, such as soft actor-critic [5], [6] and deep deterministic policy gradient [7] have also been used to learn locomotion tasks while achieving stable walking patterns at fast learning rates. Proximal policy optimization (PPO) [8] has been used to train multiple workers at once in rich environments by synchronizing their learning experiences to maximize the overall performance [9] and to teach different musculoskeletal models to walk in the OpenSim environment [10], [11]. Table I summarized some of the core research contributions in which DRL, either implementing on-policy or off-policy methods, have been used for simulations of locomotion tasks for physics-based human and robotic models.

TABLE I: DRL algorithms for human and robotic locomotion.

Ref.	DRL Algorithm	Application(s)
[1] (2020)	on-policy	hexapod robot
[2] (2019)	Q-learning with separate policies for exploration and exploitation (off-policy)	NAO humanoid
[3] (2017), [4] (2016)	off-policy and on-policy	multi-legged robots, humanoids
[5] (2018)	soft actor-critic with stochastic gradient descent (off-policy)	humanoid robot
[6] (2019)	soft actor-critic with entropy scaling over time (off-policy)	quadruped robot
[7] (2019)	deep deterministic policy gradient (off-policy)	multi-legged robots
[8] (2017)	PPO (on-policy)	humanoid robot
[9] (2017)	distributed PPO (on-policy)	multi-legged robots, humanoids
[10] (2019), [11] 2021	PPO (on-policy)	musculoskeletal human model

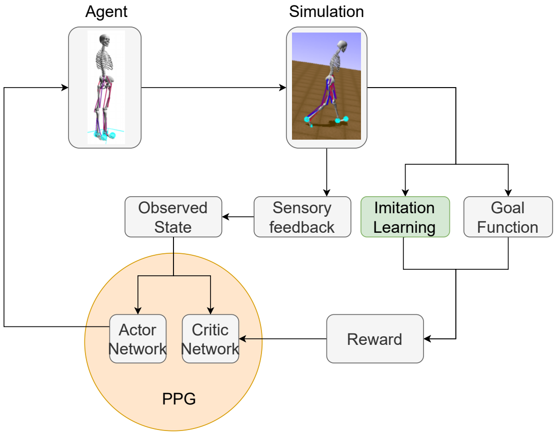


Fig. 1: An overview of the DRL architecture used in this study.

This paper focuses on DRL for the simulation of physics-based musculoskeletal models of both healthy subjects and impaired subjects (transfemoral amputee) during level-ground walking at normal speed. The DRL algorithm is inspired by the phasic policy gradient (PPG) optimization presented in [12] and is combined with imitation learning to guarantee a natural walking gait for both models. Specifically, the proposed PPG optimization, which is made of an actor network and a critic network, includes an auxiliary learning phase in the training of the actor network that accounts for the pelvis velocity and the muscles activations. Figure 1 shows an overview of the proposed architecture for teaching the agent (either the healthy subject or the transfemoral amputee) to generate normal level-ground gait patterns in the open source simulation environment Open-

This work was funded by the European Commission's Horizon 2020 Programme as a part of the project MyLeg under grant no. 780871.

The authors are with the Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, Faculty of Science and Engineering, University of Groningen, The Netherlands. stratosmitaros@gmail.com, {v.raveendranathan, r.carloni}@rug.nl

Sim [13]. The agent is trained by the DRL with PPG optimization, which receives a reward (computed on an objective function and an imitation learning term) and the observed muscles' and joints' states of the agent as inputs, and outputs an action, i.e., the activation of the muscles.

To summarize, the contributions of this paper are: (i) To propose the use of an on-policy DRL method with off-policy characteristics (i.e., PPG with imitation learning) for physics-based musculoskeletal models simulations in OpenSim; (ii) To modify the state of the art implementation of PPG to account for the musculoskeletal model; (iii) To analyze and evaluate the kinematic data and muscle forces of a healthy and an impaired model while walking on a level-ground terrain at normal speed; (iv) To compare the performances of PPG with imitation learning (on-policy method with off-policy characteristics) with our previous work on PPO with imitation learning (on-policy method) [11], and to show the improved performances of the former.

The remainder of the paper is organized as follows. The two musculoskeletal models are presented in Section II together with the public experimental data-set used for imitation learning and validation. Section II describes the method proposed in this study, with details about the deep neural network, the PPG optimizer, and the reward function. Section IV presents and discusses the results obtained during simulations. Finally, concluding remarks are drawn in Section V.

II. Materials

This Section presents the two physics-based musculoskeletal models used in this study, i.e., the healthy and the impaired (transfemoral amputee) models, and the public data-set used for the imitation learning and the validation of the method.

A. *The Models*

The two physics-based musculoskeletal models have been developed in OpenSim and are shown in Figure 2. The healthy model, as proposed in [14], consists of a total of 18 healthy Hill-type muscles (9 per leg) to control 10 degrees of freedom. The impaired (transfemoral amputee) model, as proposed in our previous work [11], consists of 19 healthy Hill-type muscles (11 in the healthy left leg and 8 in the amputated right leg) to control 12 degrees of freedom. Specifically, the impaired reduced-muscles right leg has two additional degrees of freedom (one per leg) for the hip adduction/abduction, and only uni-articular muscles (i.e., two agonist/antagonist muscles at the hip joint, two agonist/antagonist muscles at the knee joint, and two agonist/antagonist muscles at the ankle joint). This means that the bi-articular muscles (i.e., rectus femoris and gastrocnemius) have been removed from the right leg.

B. *Imitation/Validation Data-set*

The data-set used for the imitation learning and for the validation of the DRL algorithm belong to a public data-set [15]. The data was collected on 83 typically developing children by measuring the kinematics and kinetics of the hip, knee, and ankle joints, the surface electromyographic signals, and the spatio-temporal data. This study uses the pelvis, hip, knee, and ankle joints' angles, velocities, and the ground reaction forces.

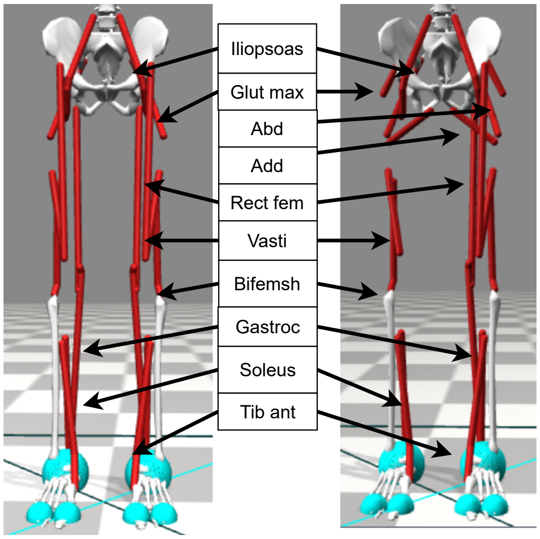


Fig. 2: The healthy model (left) [14] and the transfemoral amputee model (right) [11].

III. Method

This study proposes to use DRL to teach physics-based musculoskeletal models of both healthy and impaired (transfemoral amputee) subjects to walk on a level ground terrain at normal speed. The DRL algorithm builds upon our previous work [11], where we used the PPO optimization algorithm in combination with imitation learning, and introduces the PPG optimization algorithm in combination with imitation learning. Specifically, the PPG optimization algorithm is inspired by the work in [12] and, in this study, it is specialized for the physics-based musculoskeletal models under investigation and has been modified in the auxiliary phase.

A. *Deep Neural Networks: Actor and Critic Networks*

As in conventional actor-critic algorithms, this study uses an actor network and a critic network. Each deep neural network is a feed-forward artificial neural network, i.e., a multi-layer perceptron, composed of 4 layers, as in [11]. Specifically, the input layer depends on the observation space of the model, i.e., 218 nodes for the healthy model and 221 for the transfemoral amputee model. Afterwards, there are 2 hidden layers of 312 nodes each. Finally, the output layer of the actor network has 19 nodes (healthy model) or 20 nodes (impaired model), which includes the muscles activations (18 for the healthy model and 19 for the impaired model) and the value V_{θ_π} of the policy for the parameters θ_π. The output layer of the critic network has only one node, i.e., the value V_{θ_V} for the parameters θ_V.

B. Optimization Algorithm

1) PPO: Let $\pi_{\theta_{old}}(a_t|s_t)$ be the old policy and $\pi_\theta(a_t|s_t)$ the new policy, where θ and θ_{old} are the new and old parameters, and a_t and s_t are the action vector and the state vector at time step t, respectively. The ratio between the probabilities of the new and the old policy at time step t is:

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \tag{1}$$

To update the policy, PPO uses the following loss function [8]:

$$L^{CLIP}(\theta) = \mathbb{E}_t\left[min(r_t(\theta)A_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t)\right] \tag{2}$$

where $\mathbb{E}$ is the expected value, A_t is the advantage estimation, i.e., the difference between the expected and the real reward from an action, and ϵ is the clip value. If the probability ratio falls outside the range $[(1-\epsilon), \cdots, (1+\epsilon)]$, the advantage function is clipped to prevent too large policy updates.

To update the value, the mean square error loss is used, i.e.:

$$\frac{1}{2}(V_\theta(s_t) - V_{target})^2 \tag{3}$$

where $V_\theta(s_t)$ is the value of the policy for the parameters θ and V_{target} is the value returned by the environment due to the action a_t.

2) PPG: To reduce interference, in PPG, a separation between the training of the policy and the value function has been introduced [12], which is achieved by using two separate networks, i.e., the actor network which yields both the policy and the value, thus benefiting from parameter sharing like PPO does, and the critic network which yields the value. The two networks are trained separately. Specifically, the actor network has two loss functions. The first loss function, which is used for N_π (equals to 64 as in [12]) iterations during the policy phase, is:

$$L^{CLIP}(\theta_\pi) = \mathbb{E}_t\left[min(r_t(\theta_\pi)A_t, clip(r_t(\theta_\pi), 1-\epsilon, 1+\epsilon)A_t)\right] \tag{4}$$

where $r_t(\theta_\pi) = \frac{\pi_{\theta_\pi}(a_t|s_t)}{\pi_{\theta_{\pi_{old}}}(a_t|s_t)}$ is the probability ratio between the probabilities of the new and the old policy at time step t in the parameters θ_π and $\theta_{\pi_{old}}$, respectively.

The second loss function, which is used during the auxiliary phase (optimized for 6 epochs), is:

$$\begin{aligned} L^{JOINT} &= L^{KL} + L^{AUX} \\ &= \alpha\mathbb{E}_t[KL[\pi_{\theta_{\pi_{old}}}(\bullet|s)||\pi_{\theta_\pi}(\bullet|s)]]+ \\ &\quad +\tfrac{1}{2}\mathbb{E}_t(V_{\theta_\pi}(s_t) - V_{target})^2 \end{aligned} \tag{5}$$

where KL indicates the Kullback-Leiber divergence (which measures how one probability distribution is different from another) and α (which is set to 1 as in [12]) is a hyperparameter controlling the distance between the two policies.

The frequency of the auxiliary phase does not need to be too regular, since it interferes with the policy optimization. It therefore happens infrequently between the clip optimizations, or policy phase according to [12]. During those intermediate updates, an experience buffer is filled with experiences. By doing so, PPG trains both with policies from the current running trajectories, as well as with varying policies from previously encountered experiences.

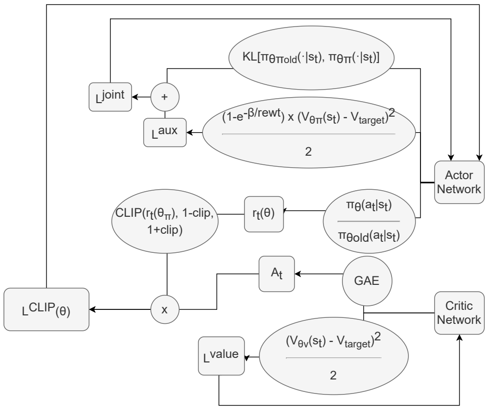


Fig. 3: Detailed diagram of the modified PPG optimization algorithm as proposed in this study.

For the critic network, the mean square error loss is used to update the value, i.e.:

$$\frac{1}{2}(V_{\theta_v}(s_t) - V_{target})^2 \tag{6}$$

where $V_{\theta_v}(s_t)$ is the value of the critic network for the parameters θ_v and V_{target} is the value returned by the environment due to the action a_t.

3) Proposed Modified PPG: In the proposed modified PPG, the two networks, i.e., the actor network which yields both the policy and the value, and the critic network which yields the value, are still trained separately. The actor network has two loss functions: the first one is also given by Equation 4 and is used during the policy phase. The hyperparameters controlling the clip have been set as follows: ϵ = 0.2 and the entropy coefficient is 0.01 to allow for balanced exploration.

The second loss function has been modified with respect to Equation 5 and is given by:

$$\begin{aligned} L^{JOINT} &= L^{KL} + L^{AUX}_{mod} \\ &= \alpha\mathbb{E}_t[KL[\pi_{\theta_{\pi_{old}}}(\bullet|s)||\pi_{\theta_\pi}(\bullet|s)]]+ \\ &\quad +\tfrac{1}{2}\mathbb{E}_t(1-e^{-\frac{\beta}{rew_t}})(V_{\theta_\pi}(s_t) - V_{target})^2 \end{aligned} \tag{7}$$

where rew_t is the reward returned by the environment (based on the pelvis velocity and the muscles activations) and γ is a a scaling factor, which is set to 1.2 $\cdot 10^{-3}$.

For the critic network, the same mean square error loss as in PPG (see Equation 6) is used to update the value.

Figure 3 shows a detailed diagram of the modified PPG optimization algorithm as proposed in this study.

C. Reward Function

The action generated by the model results in a reward, which can be either positive or negative depending on the action the model just took. If the model does a favorable action, it receives a positive reward. If the model does an unfavorable action, it receives a negative reward, i.e., a penalty. To learn the tasks, the model should receive as much reward possible.

In this paper, the reward function is calculated at each time step t and consists of four parts, i.e.:

$$J_{goal}(\pi)_t = \sum_t (r_{distance} - p_{velocity} - p_{costs}) + \sum_t (r_{alive}) \tag{8}$$

where $r_{distance}$ is the reward based on the distance that the model's pelvis covered since the last timestep, $p_{velocity}$ is the penalty for deviating too much from the desired velocity (i.e., 0.75 m/s, 1.25 m/s, or 1.75 m/s, depending on which one it matches the closest), p_{costs} is the penalty on the muscles' fatigue, and r_{alive} is the reward for the amount of timesteps spent without falling. The model is considered alive as long the pelvis is 0.75 m above ground level.

Finally, the reward function is as follows [11]:

$$J(\pi)_t = 0.4 \cdot J_{goal}(\pi)_t + 0.6 \cdot J_{imitation}(\pi)_t \tag{9}$$

where the added imitation term $J_{imitation}(\pi)_t$ uses the experimental data to ensure that the algorithm converges to a solution and that the model develops a natural gait pattern.

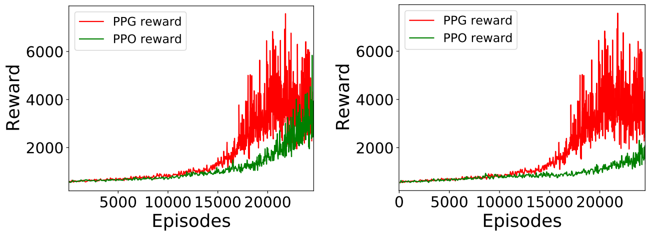
IV. Results and Discussion

This Section presents the results obtained for both the healthy and impaired (transfemoral amputee) models when the modified PPG with imitation learning is used, and compares its performances to PPO with imitation learning [11]. Specifically, the kinematic data and the fiber forces of some muscles, extracted from the two models against the experimental data, are reported and discussed.

A. Reward

Figure 4a shows the rewards received by PPG (red) compared to PPO (green) for the healthy model. It can be noted that PPG finds an optimal reward policy sooner than PPO. Specifically, PPG starts learning a policy from ∼15.000 episodes and the reward increases at a steady pace and maxes out a little over PPO. PPG takes ∼10.000 episodes less to reach similar a reward as well as surpass PPO. A similar performance is achieved on the transfemoral amputee model on which an optimal policy is found at ∼15.000 episodes, as shown in Figure 4b.

Table II reports the total reward (mean and standard deviation) received by the two models with PPO and PPG algorithms. For the healthy model, the mean reward with PPG is almost 1.5 times greater than with PPO. This increase to the mean reward comes with the price of a higher standard deviation, which also increased by a factor of 1.25. For the transfemoral amputee model, the mean reward with PPG is almost 2 times greater than with PPO, while the standard deviation has increased by a factor of 1.5. The general observation is that within the same number of episodes for both algorithms, when using the same model, the proposed PPG algorithm finds an optimal policy faster than PPO.



(a) Healthy model

(b) Transfemoral amputee model

Fig. 4: Reward per episode obtained with PPG (red) and PPO (green) for the healthy (a) and transfemoral amputee models (b).

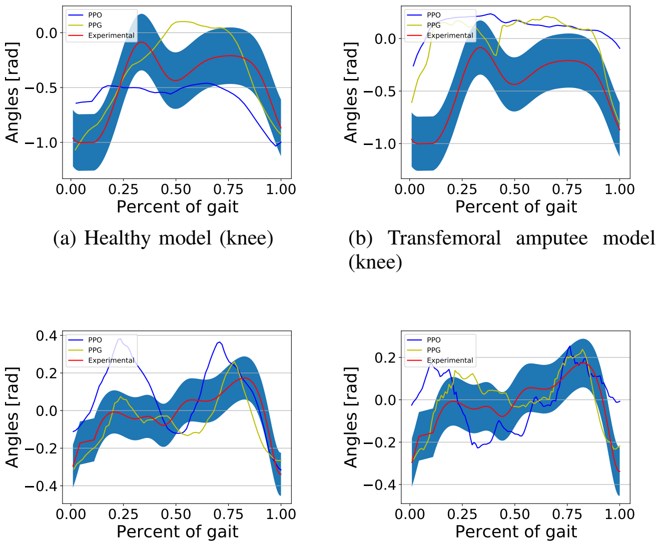
TABLE II: Total reward (mean and standard deviation) received by the two models with PPO and PPG.

	PPG		PPO	
	mean	std	mean	std
Healthy	3913.29	1781.23	2724.78	1444.45
Transfemoral Amputee	3603.93	1733.40	1797.91	1105.64

B. Kinematic Data

Figure 5 shows the angular positions of the knee and ankle joints for both the healthy and transfemoral amputee models during one gait cycle, obtained with PPG (green) and PPO (blue), and overlaid on the experimental data (shaded blue area with the mean value in the red line). From Figure 5a, it can be noted that the knee angles in the healthy model are fairly accurate with respect to the experimental data when simulated with PPG, and better than PPO. However, in the transfemoral amputee model, due to the absence of the bi-articular muscle (i.e., gastrocnemius), the knee flexion is highly affected and the kinematic data are not accurate with respect to the experimental data, as shown in Figure 5b. Interestingly, thanks to the presence of the vasti and its high utilization even in early stance because of the absence of bi-articulation, the knee extension is still possible. In the figure, it can also be observed that, around 30% of the gait cycle, PPG is able to recruit the knee flexors to initiate toe-off for the swing phase, which is on the contrary not achieved by PPO. From Figure 5c, it can be noted that also the ankle angles in the healthy model are fairly accurate with respect to the experimental data when simulated with PPG, and better than PPO. PPG can recruit the ankle plantarflexors and dorsiflexors in a similar trend as the experimental data. The same can be concluded from Figure 5d for the transfemoral amputee model thanks to the presence of the major muscles (i.e., the soleus and tibialis anterior) for plantarflexion and dorsiflexion.

To quantify the closeness between the simulated and the experimental data of the knee and ankle joints for both models, z-scores have been calculated and reported in Table III. The z-scores confirm that the simulated knee and, especially, the ankle data have a more faithful shape to the experimental data when PPG is used instead of PPO.



(a) Healthy model (knee)

(b) Transfemoral amputee model (knee)

(c) Healthy model (ankle)

(d) Transfemoral amputee model (ankle)

Fig. 5: Kinematic data (angular positions of the knee and ankle joints) obtained with PPG (green) and PPO (blue) for the healthy model (left) and for the transfemoral amputee model (right) plotted against the experimental data (shaded blue area with the mean value in the red line).

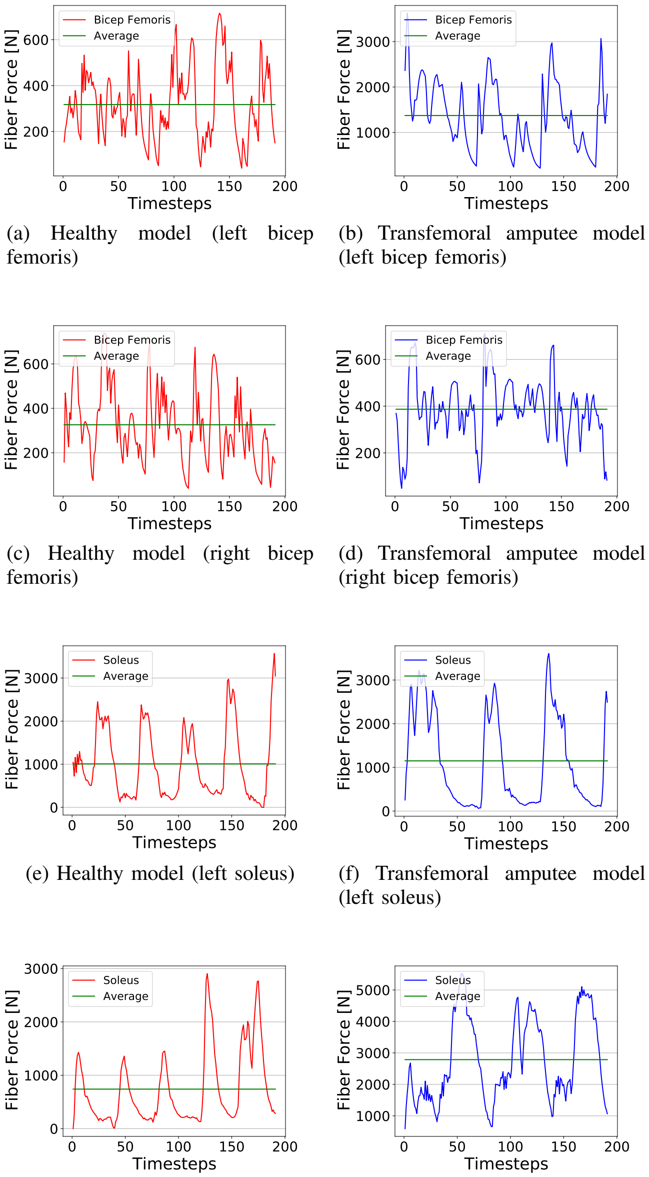
TABLE III: Z-scores between the simulated (with PPG and PPO) and the experimental angular positions of the knee and ankle joints for the healthy and transfemoral amputee models.

	PPG		PPO	
	knee	ankle	knee	ankle
Healthy	-1.18	0.25	-1.42	0.56
Transfemoral Amputee	-1.81	0.36	-2.76	0.66

C. *Muscle Fiber Forces*

Figures 6 and 7 show the fiber forces of the bicep femoris, soleus and of the vasti and tibialis anterior muscles, respectively, for both legs of the healthy and the transfemoral amputee models over approximately three gait cycles. The green lines denote the mean fiber force over the time period. By comparing Figures 6g and 6h, it can be noted that to compensate for the loss of the gastrocnemius muscle in the transfemoral amputee model, the soleus muscle on the right leg of transfemoral amputee model is activated the most. Similarly, by comparing Figures 6c and 6d, it can be noted that to compensate the loss of rectus femoris muscle in the transfemoral amputee model, the biceps femoris model has to exert much higher fiber forces. The peak muscle fiber force in the biceps femoris is five times higher than the peak of healthy model counterpart. Additionally, by comparing Figures 7c and 7d, it can be observed that also the vasti muscle has a significant increase in the average muscle force (i.e., 1000 N).

Table IV reports the mean fiber forces and the difference between the left and right leg. It can be noted that the healthy model has comparatively lower difference for the mean fiber forces of the left and right leg than the transfemoral amputee model. This difference explains the asymmetry in the gait of the transfemoral amputee model, which is due to the absence of the bi-articular muscles (i.e., the bicep femoris and gastrocnemius). Moreover, the higher fiber forces for the vasti and soleus for the transfemoral amputee model can be observed in an increase in the difference between the left and right leg.



(a) Healthy model (left bicep femoris)

(b) Transfemoral amputee model (left bicep femoris)

(c) Healthy model (right bicep femoris)

(d) Transfemoral amputee model (right bicep femoris)

(e) Healthy model (left soleus)

(f) Transfemoral amputee model (left soleus)

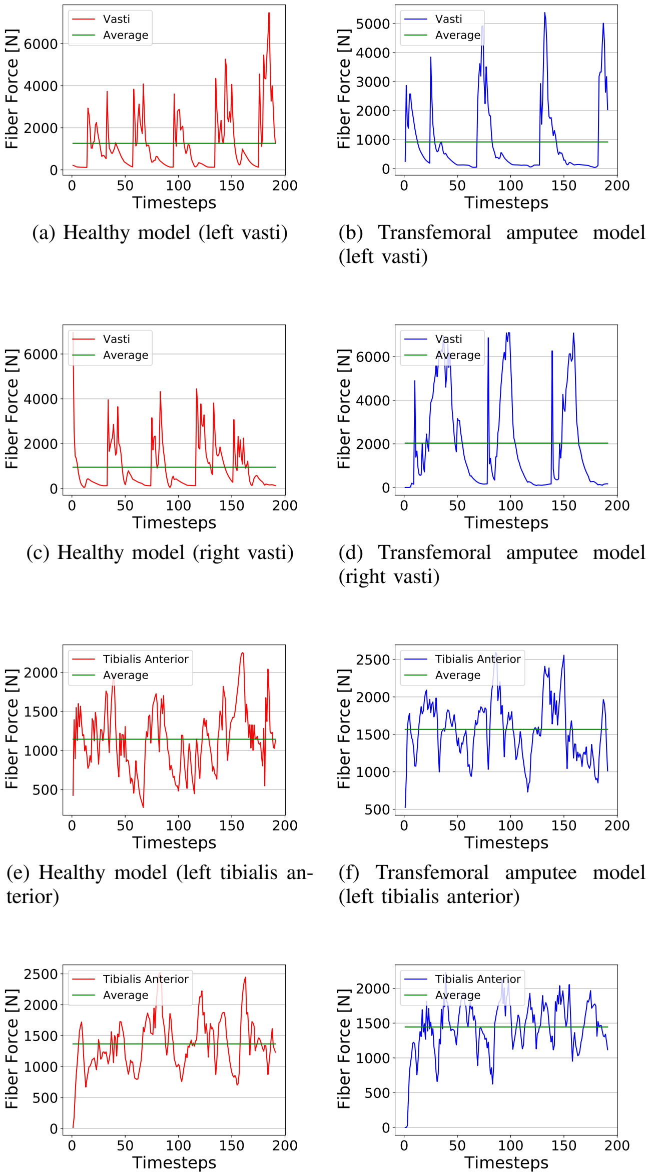
(g) Healthy model (right soleus)

(h) Transfemoral amputee model (right soleus)

Fig. 6: Fiber forces of the bicep femoris and soleus muscles for both legs of the healthy (red) and transfemoral (blue) models.

V. Conclusions

This paper proposed to use of DRL for teaching physics-based musculoskeletal models to walk on level ground



(a) Healthy model (left vasti)

(b) Transfemoral amputee model (left vasti)

(c) Healthy model (right vasti)

(d) Transfemoral amputee model (right vasti)

(e) Healthy model (left tibialis anterior)

(f) Transfemoral amputee model (left tibialis anterior)

(g) Healthy model (right tibialis anterior)

(h) Transfemoral amputee model (right tibialis anterior)

Fig. 7: Fiber forces of the vasti and tibialis anterior muscles for both legs of the healthy (red) and transfemoral (blue) models.

TABLE IV: Results for the comparison of the difference in muscle usage between the healthy and transfemoral amputee models.

		Healthy		
	Muscle	Left	Right	Δ
Activation mean	Bifemsh	0.68	0.63	0.051
	Vasti	0.21	0.19	0.022
	Soleus	0.23	0.21	0.036
	Tibialis Ant.	0.63	0.73	0.10

		Healthy		
	Muscle	Left	Right	Δ
Fiber Force mean	Bifemsh	370.41	330.43	39.97
	Vasti	1326.39	1112.01	214.39
	Soleus	1047.047	822.93	224.10
	Tibialis Ant.	1037.25	1242.90	205.65

		Transfemoral Amputee		
	Muscle	Left	Right	Δ
Activation mean	Bifemsh	0.64	0.67	0.03
	Vasti	0.07	0.19	0.12
	Soleus	0.33	0.51	0.18
	Tibialis Ant.	0.75	0.69	0.05

		Transfemoral Amputee		
	Muscle	Left	Right	Δ
Fiber Force mean	Bifemsh	346.20	341.07	5.12
	Vasti	384.47	1107.08	722.61
	Soleus	1194.97	2466.54	1271.57
	Tibialis Ant.	1368.31	1202.18	166.12

terrains at normal speed. The proposed DRL algorithm uses a modified PPG optimization algorithm combined with imitation learning.

The results show that the modified PPG with imitation learning outperforms PPO with imitation learning in terms of overall faster learning rate, in more consistent gait patterns, and closeness to the experimental data. Differences between the two algorithms also include an increased, yet more realistic, use of the knee joint in the transfemoral amputee model.

This research shows that the simulation of physics-based models can take advantage of the deep reinforcement learning and, specifically, of an actor-critic network structure, to develop walking patterns. In fact, the joint angles yielded by the simulations are close to the used experimental dataset, where the transfemoral amputee model showed that the vasti and soleus muscles would need higher activations to compensate for the lack of the bi-articular muscles to generate the gait.

References

[1] T. Azayev and K. Zimmerman, "Blind hexapod locomotion in complex terrain with gait adaptation using deep reinforcement learning and classification," *Journal of Intelligent & Robotic Systems*, vol. 99, pp. 659–671, 09 2020.

[2] C. R. Gil, H. Calvo, and H. Sossa, "Learning an efficient gait cycle of a biped robot based on reinforcement learning and artificial neural networks," *MDPI Applied Sciences*, vol. 9, no. 502, 2019.

[3] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," in *Deep Reinforcement Learning Symposium, International Conference on Neural Information Processing Systems*, 2017.

[4] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *International Conference on Machine Learning*, 2016.

[5] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International Conference on Machine Learning*, 2018.

[6] T. Haarnoja, S. Ha, A. Zhou, J. Tan, G. Tucker, and S. Levine, "Learning to walk via deep reinforcement learning," in *Robotics: Science and Systems*, 2019.

[7] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," in *International Conference on Learning Representations*, 2016.

[8] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *OpenAI*, 2017.

[9] N. Heess *et al.*, "Emergence of locomotion behaviours in rich environments," *arXiv:1707.02286*, 2017.

[10] L. Kidzinski, C. Ong, S. Mohanty, J. Hicks, S. Carroll, B. Zhou, and W. Jaskowski, "Artificial intelligence for prosthetics: Challenge solutions." *The NeurIPS'18 Competition: From Machine Learning to Intelligent Conversations*, vol. 69, pp. 1–49, 2019.

[11] L. de Vree and R. Carloni, "Deep reinforcement learning for physics-based musculoskeletal simulations of healthy subjects and trans-femoral prostheses' users during normal walking," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 607–618, 2021.

[12] K. Cobbe, J. Hilton, O. Klimov, and J. Schulman, "Phasic policy gradient," in *International Conference on Machine Learning*, 2021.

[13] S. Delp *et al.*, "Opensim: Open-source software to create and analyze dynamic simulations of movement," *IEEE Transactions on Biomedical Engineering*, vol. 55, pp. 1940–50, 2007.

[14] L. Kidzinski, S. Mohanty, C. Ong, J. Hicks, S. Francis, S. Levine, M. Salathe, and S. Delp, "Learning to run challenge: Synthesizing physiologically accurate motion using deep reinforcement learning," in *NIPS 2017 Competition Book*. Springer, 2018.

[15] M. Schwartz, Z. Rozumalski, and J. Trost, "The effect of walking speed on the gait of typically developing children," *Journal of Biomechanics*, vol. 41, pp. 1639–1650, 2008.