Deep Reinforcement Learning for Gait generation of a Musculoskeletal Model in a Physics-Based Simulation on Even and Uneven Terrain.

Bachelor's Project Thesis

Rutger Luinge, s4013484, r.luinge@student.rug.nl,
Supervisor: Raffaella Carloni

Abstract: This research focused on the use of a deep reinforcement learning algorithm to train a musculoskeletal model to walk on even and uneven terrain. To achieve this goal a learning agent was used, which we were able to teach based on the agent his past experiences, the objective of this agent was to generate a healthy gait pattern. This goal was achieved by using proximal policy optimization in combination with imitation learning. The data used in this research consists of the joint positions and velocities of a healthy walking subject during a walking trial, taken from an open source data set. This data was scaled to the model, which consisted of 22 muscles and 14 degrees of freedom. The different muscles in the musculoskeletal model could be actuated using the open source software OpenSim4.3. By using this software and the described learning architecture, the model was taught to walk on even terrain. The model/policy was subsequently transferred to uneven terrain to test the stability and capabilities of the architecture, and to simulate a more realistic scenario. As a result the model was able to walk and show a healthy gait cycle. When the model got transferred to the uneven terrain, the agent was still able to perform gait cycles but on the other hand the gait showed less stability throug not staying upright after the first two gaits. At the cost of knee flexion the algorithm was able to make the agent walk on the uneven terrain. The left side of the musculoskeletal model seemed to show less performance than the right side when we compared these results to the corresponding imitation data. We could conclude that the proximal policy optimization algorithm can generate a healthy gait cycle within the simulation. The learning architecture is still in need of improvements, focusing on the lagging left side of the body and generating a more supple gait when the policy is transferred to the uneven terrain.

1 Introduction

In this study we are using deep reinforcement learning to get a better understanding of the human gait cycle. In the simulation we use a musculoskeletal model similar to a healthy human body to study a walking trial with the goal to generate a healthy gait pattern. A traditional musculoskeletal simulates a human body, by replicating the bones, muscles and joints. This study, which is a continuation of the work by de Vree [7], incorporates a more advanced model containing 22 muscles, ultimately to improve the generated gait in comparison with the 18 muscle model. The knowledge gained with this study benefits the MyLeg project [1]. MyLeg will design and produce transfemoral prosthetic legs which can be intuitively operated in numerous daily tasks. Therefore the importance of this research lies in finding a good reinforcement learning policy to support these intelligent prosthetic

legs. To make sure that these intelligent prosthesis are capable of adapting to the environment, the musculoskeletal model is enhanced with additional muscles, and freedoms to move in. In this research we will focus on normal ground-level walking and eventually transfer the walking on a rough terrain mesh [6]. A Deep Reinforcement Learning (DRL) algorithm is used to try to generate a healthy gait pattern for the model. A DRL algorithm is a combination of reinforcement learning (RL) and a neural network: in this research a multi-layer perceptron (MLP) neural network is used. In RL, an agent or model is rewarded for each of its actions in a current state of the agent based on the events that the agent experienced. In the case of this study, the position and velocity of the musculoskeletal model was taken as the state. A model or agent will gain a higher reward for desired actions and vice versa. DRL is a very popular strategy for the usage of gait analysis [20; 21; 2] and musculoskeletal simulation. A more optimized DRL algorithm was used in this study, named: Proximal Policy Optimization [18]. The algorithm is used to optimize the policy, which is able to contract the muscles to control the musculoskeletal model, by limiting the allowed changes in generating the new policy, causing the risk of generating a worse policy to reduce. In combination with PPO, the policy was trained using imitation learning [11; 26]. This combination of PPO and imitation learning has shown significant improvement in training [7; 11]. After training the agent to walk on normal ground-level walking, the policy was transferred to the rough terrain, to analyse the capabilities of the trained policy. While PPO has already shown results on uneven terrain using 8 degrees of freedoms (DOF) [24], using 14 degrees of freedom is much more complex due to the extra dimensions the model needs to be controlled in. Transferring the policy to uneven terrain should result in more muscle activation and an increase in hip, knee and ankle work [22]. After training on level-ground the gait cycles of the model [13] of both terrain meshes will be compared to study the efficiency and adaptability of the trained policy. In addition, the increase in muscle activity and force will be compared to the expected behavior [22] when transferring to uneven terrain. Besides comparing the differences between the different terrains, a comparison and analysis on the joint angles, muscle activations, and muscle forces will be distinguished between the imitation data, the learned policy on even, and uneven terrain.

In this study we have three main goals. First of all, this research has ran simulations on the new, more profound musculoskeletal model [15], which consists of 22 muscles and 14 DOF. The goal of this first task is to check if the results [7] can be replicated in this new environment and result in a policy which can achieve stable walking, thus, being able to corrects itself when it is out of position and generate a comparable gait cycle to the imitation data. This increase in degrees of freedom gives the agent the possibility to actuate muscle groups to move sideways in the z−direction, which makes the domain of movement significantly more complex. The second task is to generate gait on uneven-terrain. This task is achieved by transferring the trained policy of the architecture to the uneven terrain mesh and observing the walking behavior and the gait cycle in comparison to the level-ground walking task. Muscle activation will be monitored to see whether the uneven terrain results in more muscle activity of the antagonist muscle groups [23]. The third goal of this study is to compare the data from the even and uneven terrain with one another, whilst taking the imitation data as a control group as this is the desired behavior of the model. The gait cycles will have to be analyzed in comparison to the imitation data to give conclusions on the success of the architecture.

The main contributions of this study are:

- Generating human like dynamics when walking during a simulation by using a musculoskeletal model containing 22 muscles and 14 degrees of freedom.
- Using the model which is able to use movements in the z−plane of the model, generating more realistic examples in walking behavior.
- Testing and validating the advised architecture of PPO in combination with imitation learning for musculoskeletal walking.
- Validating and comparing the generated policy with the imitation data in the ground-level walking task.
- Transferring the learned policy to uneven terrain to visualize the adaptability and stability

of the policy, and compare the gait to the expected changes.

The remainder of this paper is organized as follows. section 2 describes the theoretical background of some algorithms and tools used in this research, which are necessary for understanding the remainder of the paper. section 3 describes the methodology, describing the data, data processing, reward structure, and how the research has been set up using the OpenSim software. The results of the research will be shown and explained in section 4. section 5 concludes the research whilst pointing out some final remarks.

2 Theoretical Background

This section describes the different tools and means used in this research which are necessary to get a better grasp of the remainder of this research. This includes the used simulation software, the DRL algorithm, the used musculoskeletal model, and some details of a healthy gait cycle.

2.1 OpenSim

In this research we use an open source software for the simulation of the musculoskeletal model. The software that we use, OpenSim [16] v4.3, is widely used in the study of human locomotion and therefore suits this research with the task to walk on even and uneven terrain. This software has shown promising results in the research of locomotion and the human gait cycle [7; 19; 13]. OpenSim4.3 allows us to easily combine our musculoskeletal model, as the software allows us to activate the different muscles, in our case 22 muscles, within a range between 0 and 1, describing the ratio of muscle activation. Muscle forces are calculated using the Hill type muscle model [10]. In combination with the DRL algorithm we can precisely control the musculoskeletal model according to the generated policy, which makes the software optimal for researching the different techniques used to generate these policies. OpenSim is furthermore a useful tool for analyzing the gait cycle, produced by the learned policy, as OpenSim has great visualization tools.

2.2 PPO Algorithm

The deep learning algorithm that is used during this research is the Proximal Policy Optimization[18] algorithm. PPO is a form of deep Q-Learning [25], an improved version of the Trust Region Policy Optimization (TRPO) [17], which is mostly used to learn non-linear problem such as a multilayer perceptrons (MLP) [9]. PPO is an *on-policy* algorithm, meaning that the agent/model acts based on the policy the agent is being teached. The big difference between PPO and TRPO is that PPO uses a clipping function (figure 2.1) to prevent the policy moving in a bad direction. This clipping function is used to prevent the policy from changing to a worse policy after one bad iteration of learning, or to prevent the policy from too large of a change when a good result happened, all to prevent the policy from changing too much. Equation 2.2 shows the main learning function of the PPO algorithm which is used in this architecture. PPO is currently widely used in reinforcement learning, especially since it has been picked up by OpenAI [3]. Another reason for using PPO as the preferred learning algorithm is the good results and useful recommendations made in previous research of musculoskeletal models in combination with PPO [12]. The research of de Vree and Carloni (2021) [7] has shown that the implementation of PPO algorithm and imitation learning was able to generate a policy for a musculoskeletal model, resulting in a healthy gait cycle for the simplified model:

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \tag{2.1}$$

Where $r_t(\theta)$ denotes the probability ratio of the new policy to occur, based on the old policy.

$$L(\theta) = \hat{E}_t[min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)] \tag{2.2}$$

Where θ is the policy parameter, r_t is the probability ratio of the policy in equation 2.1. $\hat{A}_t$ the estimated advantage at time t. ϵ denotes the clipping hyper-parameter as discussed in figure 2.1.

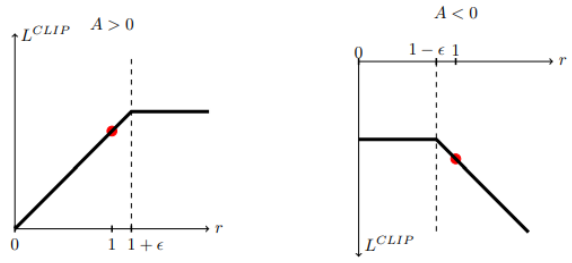


Figure 2.1: This image [18] shows how the probability ratio is being clipped. The left image shows the positive advantages (A), and the right images show the negative advantages. The probability ratio is displayed on the figures x-axis, while the L^{CLIP} is shown on the y-axis.

2.3 Imitation learning

In combination with PPO we are using imitation learning to update the policy of the system to increase the learning of the model. The work of de Vree and Carloni (2021) [7] has shown that without imitation learning, the musculoskeletal model does not achieve the task of level-ground walking. Therefore our reward shown in figure 3.1 consists of an extra imitation reward part. Imitation learning uses data from a human trial, this data is used as a reference for the agent. By calculating the error between the imitation data and the position of the agent, the architecture can use this effectively in combination with reinforcement learning.

2.4 Musculoskeletal model

The OpenSim v4.3 software can make use of musculoskeletal models. These models are a combination of different bones, joints and muscles. In the research of de Vree and Carloni [7] the model contained 18 different muscles which gave the model 9 degrees of freedom. The model used in this study, designed by A. Seth [15], aggregates 22 muscles(groups), 11 on each side of the body [15]: iliopsoas; rectus femores; vasti; tibialis anterior; soleus; gastrocnemius; bicep femoris; hamstring; gluteus maximus; hip adductors; and hip abductors, of which these last two groups were added to the 18 muscle model to increase the control in the hip movement.
The hip abductors and adductors create an extra level of complexity to the task of walking. In the research of de Vree and Carloni, the model could only be activated to move in the x (forward) and y (vertical) direction, see figure 2.3. However, in real world applications, such as the MyLeg project, we want the agent to move along in the z-direction (sideways) as well. See figure 2.3 for the used orientation plane in OpenSim. The original 9 degrees of freedom of the model are: left and right dorsiflexion, left and right knee flexion, left and right hip flexion, pelvis x− and y− position, and pelvis tilt. The increase in muscles has generated 5 additional degrees of freedom to the model: left and right hip adduction; pelvis z-position; pelvis obliquity; pelvis rotation. The differences between the 18 and the 22 muscle models can be visualized in figure 2.2 and table 2.1

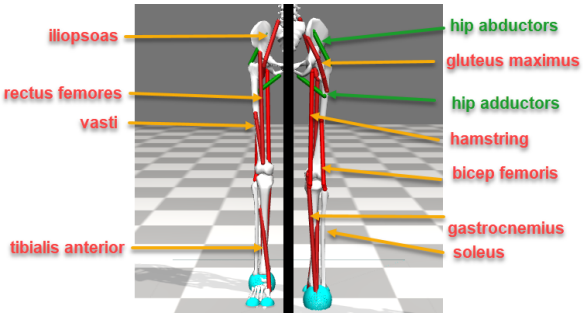


Figure 2.2: This images shows 11 different muscle (groups), these muscles are symmetric on both sides of the body resulting in 22 muscles (groups). On the left, the front view of the musculoskeletal model is visible, and on the right side, the back view is visible. The additional muscle groups (hip abductors and adductors) are colored green.

2.5 Human Gait Cycle

The human gait cycle, as shown in figure 2.4 is the normal and healthy motion of walking. The initial position of this motion is with the right leg in front of the pelvis, with no knee bending in both legs. This was also be the initial position for the DRL agent that is used in this research. During the gait the ankle flexion (dorsiflexion), knee flexion, and hip flexion will have different values throughout the cycle. These angles can differ slightly for every hu-

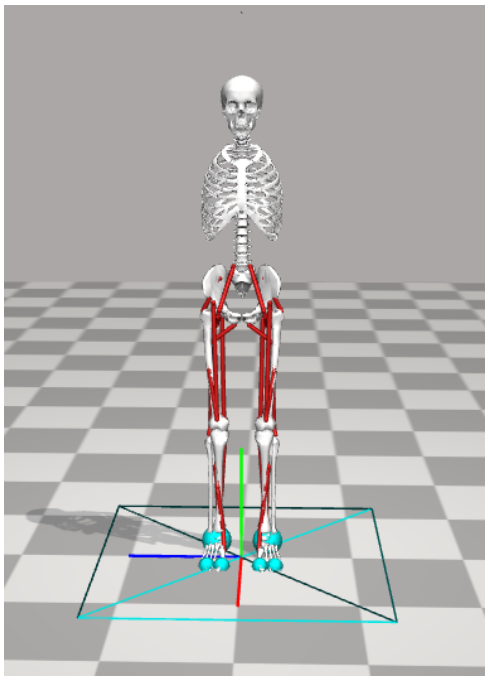


Figure 2.3: This figure shows the musculoskeletal model containing 22 muscles, the colored lines at the feet of the model describe the different orientations of the model. Where the red line is x−axis, the green line the y−axis and the blue line the z−axis. This model, unlike the 18 muscle model, is able to use to move with respect to the z−direction.

DOF	Available in	
Function	**18 muscle model**	**22 muscle model**
Hip extension	Yes	Yes
Hip ab/adduction	No	Yes
Knee flexion	Yes	Yes
Ankle flexion	Yes	Yes
Pelvis position x, y	Yes	Yes
Pelvis position z	No	Yes
Pelvis tilt	Yes	Yes
Pelvis rotation, obliquity	No	Yes
Total DOFs	**9**	**14**

Table 2.1: Comparison of the different degrees of freedom of the newly adapted model, in comparison to the 18 musculoskeletal model.

man individual and therefore could also differ a lot in imitation data. The gait cycle used from the imitation data [4] is shown as the gray area in figure 4.2. On the uneven terrain, the lower limb muscles have to work harder to reach a similar gait when walking on normal terrain [23], this results in a higher mean muscle activation in 8 lower limb muscles during walking on uneven terrain.. Previous research [23] has found an increase of the positive knee and hip work by 28% and 62% respectively and the negative knee work was increased by 26%. When comparing the results we want to find a similar increase in joint work.

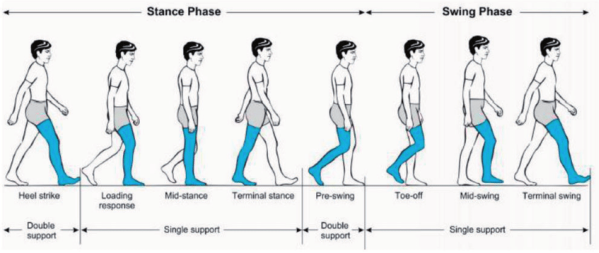


Figure 2.4: This image [14] shows the process of a healthy human gait cycle, this is considered the aim of this research in combination with transfer learning to uneven-terrain. The starting position of a trajectory is similarly initialised as the start of this gait

3 Method

This section discusses the overall architecture of the research, the data necessary for imitation learning and the processing of this data. Furthermore the reward structure is explained, and the last section explains the uneven terrain mesh as used in this study.

3.1 Architecture

The architecture used in this research is shown in figure 3.1. The musculoskeletal model/agent starts in the simulation software OpenSim4.3, from each state of the model, the reward is generated in two different parts, imitation reward and goal reward. The imitation reward is based on how similar the

position of the agent is in comparison with the imitation data at a current point in time, the goal reward is a general factor which describes if the behavior of the agent is also good, in the case of this study the agent should move forward. These rewards are combined and passed to the neural network, and used for updating the policy when the PPO batch size is reached. Other than the reward, the current observation space, thus the state of the model, is used as the input of the MLP. The observation space has a size of 124 and contains all important information about the current state of the model, table 3.1 shows that this observation state exists of: the different joints positions and velocities; significant bodies positions and velocities; pelvis orientations, velocities and acceleration; and the observation space also includes the imitation data of the current time step. This is done to give the agent some sense about the error of its current position. This addition to the observation is made as it had shown more promising results in the learning process, found by trial and error. The observation space is the input of the MLP which produces an output which is equal to 22 values between 0 and 1. These numbers represent the new muscle activations for the agent. When the muscle activations have been applied to the OpenSim software the musculoskeletal will render its new pose and the reward and observation can be gathered again. The PPO policy updates when the PPO batch size is reached. The used PPO hyperparameters, such as the batch size, can be checked out in table 3.2.

	Observation Space
Pelvis position(xyz) + velocity(xyz) + orientation (xyz) + acceleration(xyz)	4 * 3
Joint positions + velocities	5 * 2 * 2
Bodies positions(xyz) + velocities(xyz)	12 * 3 * 2
Imitation-data pose + velocity	10 * 2
Total size	**124**

Table 3.1: Observation space of the agent, and thus the input of the MLP. The observation space size is equal to 124. The observation space includes the pelvis position, velocity, orientation and acceleration. For each side of the body 5 joint (position and velocity) are taken (knee, ankle, hip flexion/adduction/rotation) and for each side (r+l) 10 significant bodies position and velocities are included. The imitation data is included as an error indication for the model.

DRL Architecture

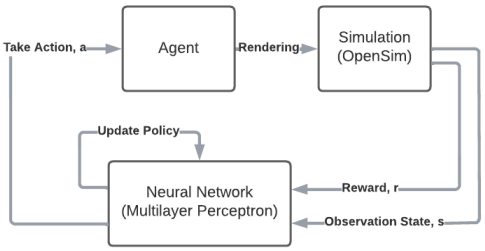


Figure 3.1: This image shows the architecture of this research including the flow of the DRL algorithm. From the simulation the reward and state are gathered and passed to the MLP, the MLP outputs an action consisting of 22 muscle activations. After the system has stored a batch of information the policy will update based on its previous experience. The cycle will continue after the new position of the model is rendered into OpenSim v4.3

3.2 Imitation Data

Imitation learning is in need of good and accurate data of a human subject, from which a robot, or in this case, an agent, can learn. This section describes the used public data set [4] and how this data has been transformed and translated to fit our musculoskeletal model. At last an inverse kinematics has been done using the scaled model, which created our final data set

3.2.1 Data Set

For the imitation learning the data from subject AB06 of the Camargo data-set is used [4]. This subject is a 20 year old male of 1.80 meters with a weight of 74.8kg. This subject was chosen to match the length and the weight of the musculoskeletal model as closely as possible. The subject had to walk on level-ground terrain, performing two 90° turns in the trajectory. The data set included the angle position/velocities of the lower limb joints at an interval of 0.005 seconds. This interval was used as the length of each time step during the learning process.

3.2.2 Data Processing

As this research is only concerned with level-ground walking in a straight line, the data had to be pre-processed in order to fit this research. Therefore we cut the first 90° turn of the data until the subject was in the desired starting pose to start the straight line walking trajectory. Furthermore we had to translate and transform the data to fit the origin of the OpenSim models initial position. We translated the data by 627 cm in x−direction (i.e., forward) and -1039 cm in the z−direction (i.e., sideways). Furthermore we rotated the data by -90° on the y−direction axis (i.e., a turn to the right). The OpenSim scale and inverse kinematics tools suffice a trace results file (TRC file), therefore our data-set had to be rearranged to fit this format.

3.2.3 Scaling and Inverse Kinematics

In order to create a data set which is usable for the OpenSim model the data had to be scaled, as differences in muscle fibers and bone lengths will otherwise result in inaccurate results, from which the agent can not learn properly. To achieve this, we used the OpenSim scaling tool, where we scaled the model to the processed data, this was done by taking a fixed position, in our case the position of the data at 12.8 seconds. As a result we got a scaled model with: a total squared error = 0.044 meters, marker error: RMS = 0.039 meters. The maximum error was found at the left heel being max = 0.086 meters. For this, an initial steady position had to be chosen (12.8 seconds). With the scaled model the final data set could be generated by running the inverse kinematics function in the OpenSim software, resulting in a level-ground walking trajectory from 12.8-16.7 seconds. Where the starting position is to be chosen to be the start of a human gait cycle.

3.3 Reward

The reward function is designed for the agent to reward positive behavior. Whenever the behavior is beneficial for the goal, the reward should be high, and whenever the behavior is not moving towards the goal, the agent had a poor set of movements which it should not learn from. This reward function is a combination of two larger factors, the goal reward, a value function based on the position and velocity of the agent, and a imitation reward which is based on the joint positions and velocity of the agent in comparison to the imitation data as described in section 3.2. Both parts of the reward use an exponential function to limit the maximum of the partial reward to 1. This made it that neither the goal nor the imitation reward could take the upper hand.

3.3.1 Goal Reward

The structure of the goal reward is shown in equation 3.1, in which the goal reward is the result of an exponential function where the velocity penalties of the x− and z−direction of the agent are summed. These penalties are calculated by squaring the differences between the measured and the desired velocity of the agent. The desired x−velocity is based on the pelvis forward speed from the imitation data. And the desired z−velocity is equal to 0 as sideways movement is not beneficial to the goal of forward level-ground walking.

$$reward_{goal} = \sum_{t=0}^{pelvis_y<0.8} e^{-8*(x_{penalty}+z_{penalty})} \quad (3.1)$$

Where we sum the reward each time step t, until the pelvis reaches below a height of 0.8 meters.

3.3.2 Imitation Reward

The imitation reward is shown in equation 3.2. The position and velocity penalty is calculated by squaring the difference of the agent's joint position and velocity in comparison to the imitation data, and summing the components of the following 11 degrees of freedom (joints): flexion of the knees, adduction and flexion of the hips, dorsiflexion (ankles), pelvis rotation, tilt, and obliquity. The position and velocity rewards are summed together in a ratio of 75% and 25%, respectively these ratios have been taken by the research of De Vree [7]. An extra penalty has been implemented to enforce knee flexion. The reward for the knee is halved whenever the knee angle differs too much from the imitation data. This helped the agent to show a more healthy knee flexion. Which is important for future imple-

mentation in a transfemoral prosthesis.

$$reward_{imi} = \sum_{t=0}^{pelvis_y<0.8} \left(0.75e^{-2\cdot\sum_n^{11} pos_{pen}} + 0.25e^{-0.1\cdot\sum_n^{11} vel_{pen}} \right) \tag{3.2}$$

Where t denotes the time step until pelvis-y gets below 0.8 meters, and n enumerates over the 11 different joints, which are described earlier.
The final ratio between goal and imitation reward is shown in equation 3.3. An additional penalty is added in order to prevent the agent from crossing its feet, as this is impossible in the real world so we want to prevent this behavior. As a result of this, the total reward, as described in equation 3.3, is halved.

$$reward_{total} = cross_{penalty} * (0.4 * reward_{goal} + 0.6 * reward_{imitation}) \tag{3.3}$$

3.4 Additional Penalties

To speed up the training and to get the desired behavior of the agent, additional penalties have been added to the architecture, first of all the episode will end whenever the pelvis y-position is below a threshold of 0.8 meters. The work of de Vree [7] had a smaller value for this threshold. This new value has been chosen to decrease training time, since the more complex model training takes more than quadruple the training time. Additionally, an extra penalty has been added to prevent the agent's feet from crossing in an way that should not be possible. The contact forces between the model and the ground are calculated using four contact points in the feet using elastic foundation forces, the bones and muscles of the model are not solid objects and therefore could in theory go trough each other, thus the position of the feet may overlap. Whenever this erroneous pattern occurs, the total reward is decreased using: $new_{reward} = 0.5 * reward$. Furthermore, a penalty is added to encourage knee flexion, this is done by decreasing the knee velocity and position by half whenever the position or angle differs more than 0.4 radians from the imitation data.

3.5 Even Terrain

The even terrain that was used is the basic OpenSim environment, in this place the ground is used as a contact point for the model. The model itself contained 2 contact point on each feet on the heel and the toes, resulting into 4 total contact points. These contact points use elastic foundation forces to calculate the reaction forces with the ground, this is realized by placing a spring on the center of all contact meshes. The goal for the even terrain is for the model to walk in a straight line after training. This is realised by training the agent with the set parameters that can be seen in table 3.2. Note that the training has done using 4 parallel workers, to reduce the training time and generate more efficient policy optimizations. The initial position of the musculoskeletal model is with the right leg in front, simulating the start of a gait cycle. After the training finished, the results of the trained policy was generated letting the agent perform muscle activations during a maximum of 5000 time steps.

Parameter	Value	Parameter	Value
Training Time	6d 23h	MLP activation function	tanh
Parallel Workers	4	PPO batch size	512
Iterations	1967	PPO Entropy Coefficient	0.01
Time Steps (0.005 s)	9.1 M	PPO γ	0.9
MLP hidden layers	2	PPO δ	0.999
MLP hidden layer size	312	Nr. epochs optimizing loss	4
MLP input size (state)	124	PPO clip ϵ	0.2

Table 3.2: The used parameters for running the trial on uneven terrain, displaying different PPO parameters: batch size; entropy; discount factor (γ); GAE parameter (δ) and other run parameters such as the design of the multilayer perceptron and the training time.

3.6 Uneven Terrain

The uneven terrain mesh is adopted from the research of de Boer [6]. The mesh consists of uneven terrain with a maximum offset of 0.06 meters from the ground.The mesh is made the open source software Blender [5]. The terrain mesh makes use of an elastic foundation contact model, to create contact forces for the agent. In figure 3.2 the rough terrain can be observed. After training on the even terrain, the policy is transferred to the uneven terrain mesh, to check whether the agent is able to generate a gait on more difficult terrain. The initial position

in this mesh is equal to the even terrain, the model is standing on even terrain at the start, to create a stable foundation before walking on the uneven terrain. To test the policy on the uneven terrain, the position and angles of the different degrees of freedom will be monitored during a maximum of 5000 time steps, similar to the run on even terrain.

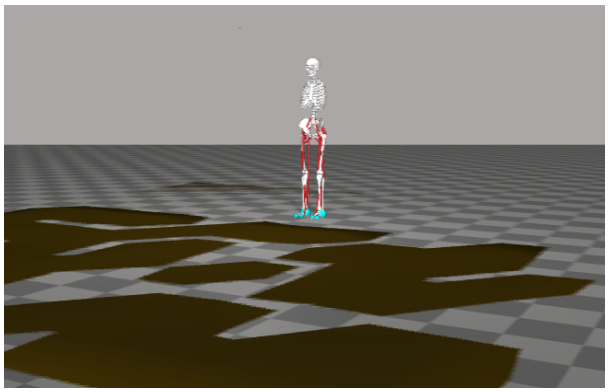


Figure 3.2: This figure shows the uneven terrain made by de Boer [6] using the open source software Blender. The image shows the terrain in the OpenSim v4.3 software. The musculoskeletal model starts at a level-ground plateau to have a stable starting position.

4 Results

This section will show the results of the DRL algorithm. The first part shows the results on the normal walking of the musculoskeletal model, comparing the results to the imitation data [4]. The second part shows the results when transferring the policy to an uneven terrain mesh, where the differences between level-ground walking will be examined and compared to the expected differences [23].

4.1 Normal Level-Ground Walking

The task of learning to walk on normal level-ground terrain is learned by the musculoskeletal model. The model achieved to stay alive ($pelvis_y > 0.8$) and move forward for 5000 cycles of the architecture, resulting in 3.35 seconds of simulation time. The walking behavior seems stable as the agent is able to keep a similar gait during the entire trajectory and it could continue if a longer trajectory is desired, however, it has to be pointed out that the individual gaits during the whole trajectory can differ, showing either a gait with no knee flexion, and sometimes showing a similar result as can be seen in figure 4.2, resulting in a healthy gait in comparison with the imitation data.

The PPO algorithms learning curves are shown in figures 4.1a, 4.1b. The curves show the mean length and reward of an episode, respectively. The green line shows the learning of the best policy for the task of walking on even terrain. The gray line shows an earlier run before the extra knee penalty had been added to generate better knee flexion. This run showed less good results in the even terrain walking but reached a longer trajectory on uneven terrain. From these graphs the learning spike in episode length and reward around episode 1100 can be noticed. At this point in time the policy could achieve a full gait cycle where the musculoskeletal model was back in a starting pose of a gait cycle, see figure 4.2. Therefore the following steps were more easily achievable for the model. The root mean squared error (RMSE) are shown in table 4.1. The results show that the RMSE is not further than 1.4 standard deviation (SD) from the imitation data. The left knee, and ankle were the only 2 joints outside of 1 SD of the imitation data (for 1 gait cycle). Which implies that the generated policy does show similarities with the imitation data. The table 4.1 shows in combination with figure 4.2 that the left side of the musculoskeletal model is less accurate than the joints located on the right side of the model as these RMSE are outside the 1 SD range. The left ankle, knee and hip joints have a 53%, 107% and 53% increase in RMSE, respectively, in comparison to the right hip joints when looking at the imitation data, in the video the difference between the left and right side of the musculoskeletal model is clearly visible. The results in figure 4.2 show a positive knee angle which at first sight should not be possible, however, this is not uncommon as previous research has also shown similar knee behavior [8], where the knee is slightly angled positively, which is the behavior of “locking” or overextending your knees. Another problem to be pointed out is the erraticness of the ankle joint shown in figure 4.2, this constant change in the ankle angle makes it hard for the model to generate

a good gait on even and especially uneven terrain [23]. In the video is clearly seen that at the end of the trajectory the agent is turning left slightly, this implies that the imitation data was not completely cut off correctly.

Joint	r/l side	RMSE	Δ l/r
ankle joint	right	8.45	4.52
	left	12.97	
knee joint	right	9.87	10.59
	left	20.46	
hip joint	right	9.38	5.01
	left	14.39	

Table 4.1: RMSE between the first gait cycle of the policy generated trajectory and the imitation data. And the difference Δ between the left and right side.

4.2 Uneven Terrain Walking

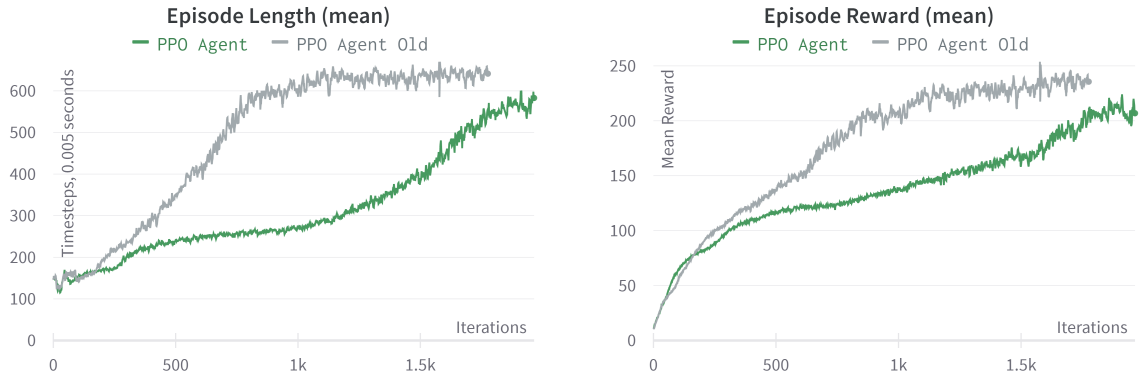
Transferring the learned policy to uneven terrain has shown a reasonable results as can be seen from figure 4.2, the result however is less stable than the even terrain. Where the policy went as long as 5000 time steps, the policy on uneven terrain only reached around ~ 3 full gait cycles. An earlier run of the architecture showed better results for the uneven terrain, reaching the end of the 5000 muscle activations when transferring the trained model. However this version stayed behind in knee flexion as it didn't include the added knee penalty because this penalty was added later to increase the performance on even terrain. The result between these different runs are visualised by 2 videos. This research has chosen to focus on using the run with the extra knee penalty (green curve in figure 4.1) as the previous runs did not show any knee flexion resulting in a unhealthy gait pattern. Other than that the importance of the knee flexion has a large factor in the manufacturing of transfemoral prostheses as these prostheses use two actuators, one of which is replicating the knee joint.
Earlier research [23] has shown that a healthy gait cycle on a more rough terrain uses more activation in 7 lower limb muscles. The results of this research did not match this increase in muscle activity as can be seen in table 4.2, where the mean muscle activations are shown. These important muscles for reaching a gait cycle all have a lower activation than the task of walking on even terrain. And thus suggest that these result on uneven terrain do not imply a healthy gait pattern. The muscle activations in figure 4.3 also seem very similar between the even and uneven terrain, implying that the policy is not transferred well and thus not adapting well to a different terrain which it is not trained on.

Muscle name	even terrain	uneven terrain
soleus	0.207	0.151
gas	0.431	0.393
hamstrings	0.388	0.341
fem	0.712	0.7
tib	0.828	0.825
vasti	0.100	0.095

Table 4.2: The mean muscle activations of the agent during $\sim$ 1 gait cycle. The uneven terrain column concerns the muscle activations for the trial with the policy transferred onto the rough terrain mesh.

5 Conclusions

The increase in complexity of the musculoskeletal model has shown generate a gait cycle for as long as the trajectory took place using the proximal policy optimization in combination with imitation learning. A gait cycle, inclusive of knee flexion, has been generated by the algorithm, which may further be improved by future research. The gait cycle seems to be healthy, even though the gait differs much between the left and right side of the musculoskeletal model. An assumption that could well be true is that this difference in performance between the left and right side of the body is caused by the scaling of the model, as the left heel of the model showed the highest error after scaling. The uneven terrain has not shown as good results as the even terrain: both the gait and RMSE are comparable across the two terrains, but the lack of stability causes the model to crash when the knee-flexion penalty had been implemented. Therefore not showing more than 2 gait cycle after collapsing and falling into the ground. Furthermore the change in muscle activations do not match with the expected change when transferring to uneven terrain, the contrary



(a) Episode length

(b) Episode Reward

Figure 4.1: The left figure shows the mean length of the trajectory of the agent. The length is in steps of 0.005 seconds as this was equal to the steps in the imitation data, as been described in section3.2. The right figure shows the mean reward of a trajectory, the reward is generated by the reward function 3.3 and summed over the whole trajectory. The X axis of both graphs shows the learning iteration of the algorithm. In both graphs, the green line annotates the best found policy for the even terrain. The gray line is an earlier run which reached 5000 timesteps when transferring to the uneven terrain, however no knee flexion was observed.

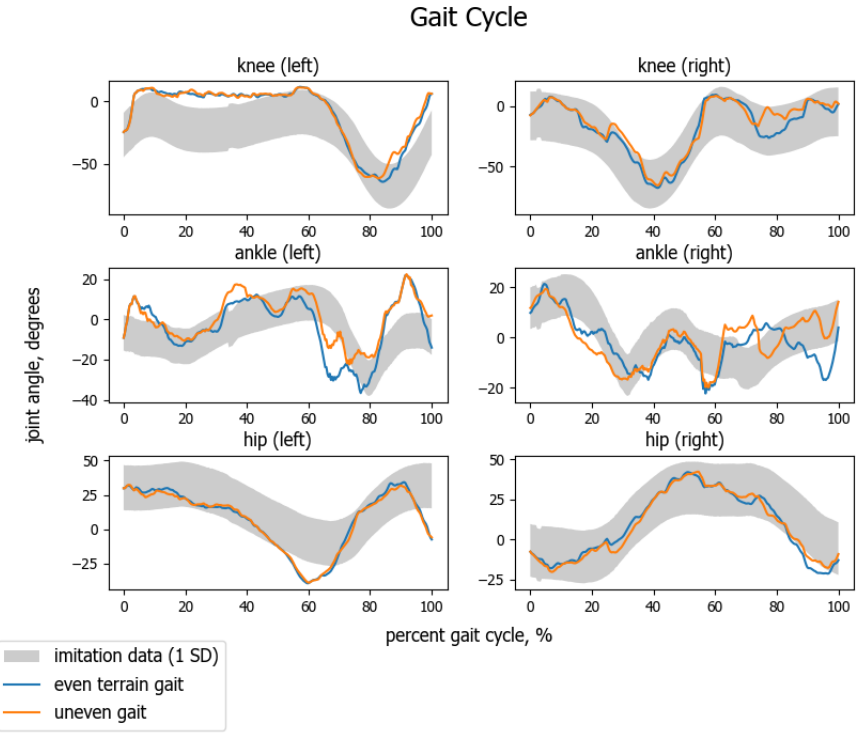


Figure 4.2: This figure shows the different joint angles during one gait cycle. The gray area shows the imitation data (1 standard deviation), the yellow line describes the gait of the agent on even terrain, and the green line is the agent transferred to uneven terrain.

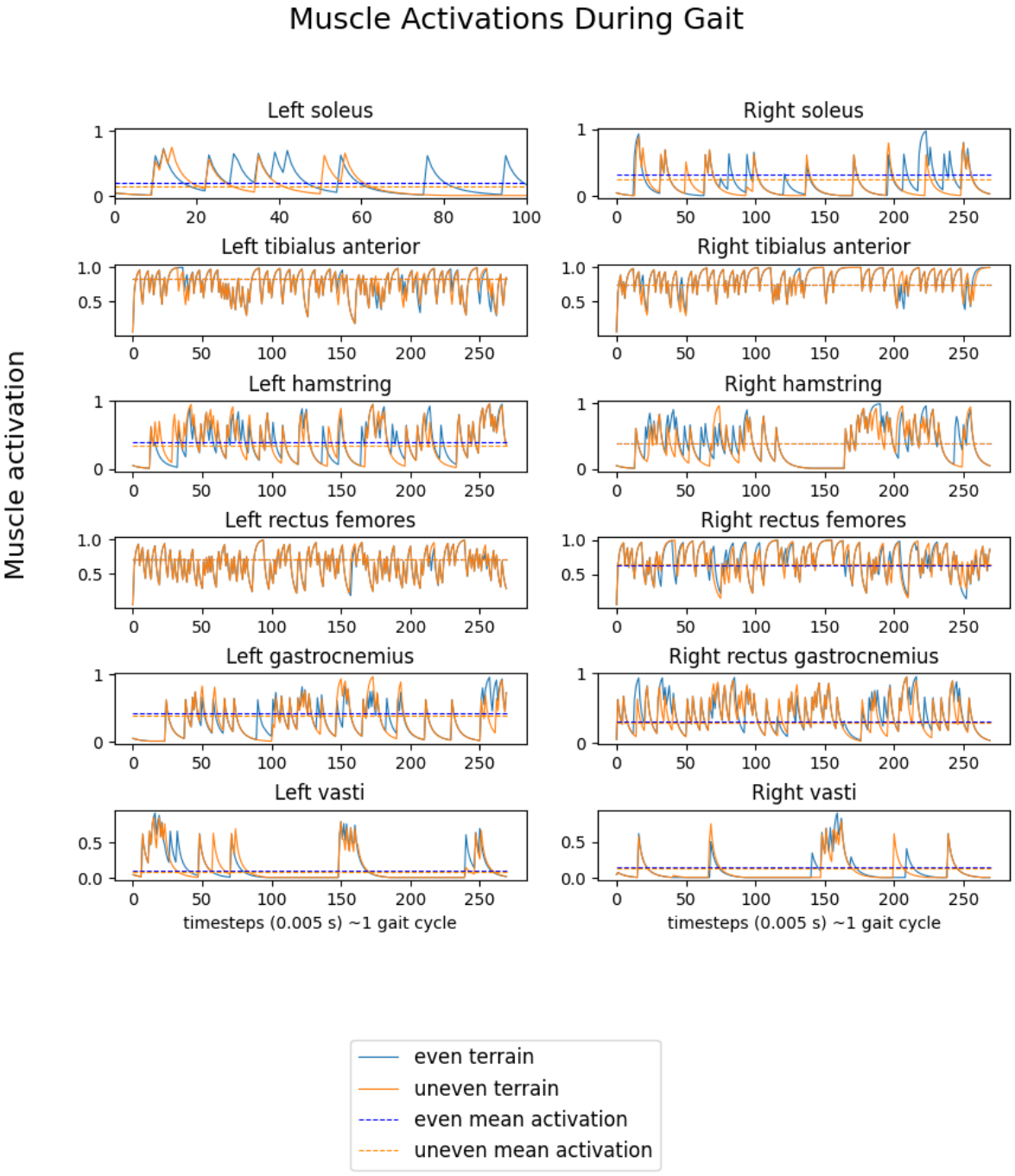


Figure 4.3: This figure shows the muscle activations during the first gait of the agent of the left and right side of the body. The blue line marks the even terrain, while the orange line shows the activations on uneven terrain. The orange striped line denotes the mean activation on the uneven terrain while the blue striped line shows the even terrain.

seemed to be true, where the activations on uneven terrain were all lower. On the other hand, with respect to the gait, without the added knee-penalty the policy was able to walk on uneven terrain and was adaptable enough to correct itself on uneven terrain with height offsets of 0.06 meters, during the whole trajectory. The main goal of this research was to make the "22 muscle, 14 degrees of freedom model" work, and this goal has been achieved. The other contributions such as making the model able to walk while it is able to move in the z−direction was also full fulfilled, and showed to be possible with the use fo the PPO algorithm in combination with imitation learning. The next step would be to go more in depth in the structure of the imitation and goal reward, some additions have already been made by this research. However, to reach a consistent healthy gait cycle, particularly for the uneven terrain, the reward structure and PPO parameters should be further analyzed. A suggestion for further research is to focus more on this reward structure, especially with respect to generating knee flexion, as this is a critical joint for a healthy gait cycle, and necessary when replacing the knee joint with an actuator to realize a "smart" transfemoral prosthesis. A second suggestion that can be made would be to focus on the complexity and the time consumption of the algorithm. Both should be decreased, as currently the training is slow and tedious. A third remark would be to revisit the scaling of the model, as this is not optimal yet and could be improved on when looking at the large scaling errors such as the heel. Finally, in line with the goals of the MyLeg project, future researchers should replace the lower limb muscles of one leg by 2 actuators to simulate a transfemoral prosthesis and try to replicate these results while using the different model.

6 Acknowledgements

The author would like to thank his supervisor, Raffaella Carloni (Professor, University of Groningen) for the supervision and helpful insights during the project, as well as the BSc students Carl Lange, Elene Poeltuijn, Robin Kock, Andrei Voinea and Massimiliano Falzari, for active discussions and recommendations to adapting the method and algorithm structure during this project.

References

[1] Myleg, Feb 2018.

[2] A. S. Anand, G. Zhao, H. Roth, and A. Seyfarth. A deep reinforcement learning based approach towards generating human walking behavior with a neuromuscular model. In *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, pages 537–543. IEEE, 2019.

[3] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[4] J. Camargo, A. Ramanathan, W. Flanagan, and A. Young. A comprehensive, open-source dataset of lower limb biomechanics in multiple conditions of stairs, ramps, and level-ground ambulation and transitions. *Journal of Biomechanics*, 119:110320, 2021.

[5] B. O. Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[6] S. de Boer. Deep reinforcement learning for physics-based musculoskeletal model of a transfemoral amputee with a prosthesis walking on uneven terrain. 2021.

[7] L. De Vree and R. Carloni. Deep reinforcement learning for physics-based musculoskeletal simulations of healthy subjects and transfemoral prostheses' users during normal walking. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 29:607–618, 2021.

[8] S. Garcia, M. Vakula, S. Holmes, D. Pamukoff, and M. Montgomery. The influence of body mass index and sex on frontal and sagittal plane knee mechanics during walking in young adults. *Gait Posture*, 83:217–22, 10 2020.

[9] M. Gardner and S. Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric Environment*, 32(14):2627–2636, 1998.

[10] A. V. Hill. The heat of shortening and the dynamic constants of muscle. *Proceedings of the Royal Society of London. Series B-Biological Sciences*, 126(843):136–195, 1938.

[11] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

[12] E. Joos, F. Péan, and O. Goksel. Reinforcement learning of musculoskeletal control from functional simulations. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 135–145. Springer, 2020.

[13] A. Nandy and P. Chakraborty. A study on human gait dynamics: modeling and simulations on opensim platform. *Multimedia Tools and Applications*, 76(20):21365–21400, 2017.

[14] N. Nordin, A. Muthalif, and M. Razali. Control of transtibial prosthetic limb with magnetorheological fluid damper by using a fuzzy pid controller. *Journal of Low Frequency Noise, Vibration and Active Control*, 37:146134841876617, 04 2018.

[15] C. Ong and A. Seth. gait14dof22musc.

[16] C. R. Raveendranathan V. Musculoskeletal model of an osseointegrated transfemoral amputee in opensim. *IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*, 2020.

[17] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1889–1897, Lille, France, 07–09 Jul 2015. PMLR.

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

[19] A. Seth, J. L. Hicks, T. K. Uchida, A. Habib, C. L. Dembia, J. J. Dunne, C. F. Ong, M. S. DeMers, A. Rajagopal, M. Millard, et al. Opensim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement. *PLoS computational biology*, 14(7):e1006223, 2018.

[20] S. Song, Ł. Kidziński, X. B. Peng, C. Ong, J. Hicks, S. Levine, C. G. Atkeson, and S. L. Delp. Deep reinforcement learning for modeling human locomotion control in neuromechanical simulation. *Journal of neuroengineering and rehabilitation*, 18(1):1–17, 2021.

[21] R. Su, F. Wu, and J. Zhao. Deep reinforcement learning method based on ddpg with simulated annealing for satellite attitude control system. In *2019 Chinese Automation Congress (CAC)*, pages 390–395, 2019.

[22] A. Voloshina, A. Kuo, M. Daley, and D. Ferris. Biomechanics and energetics of walking on uneven terrain. *The Journal of experimental biology*, 216, 08 2013.

[23] A. S. Voloshina, A. D. Kuo, M. A. Daley, and D. P. Ferris. Biomechanics and energetics of walking on uneven terrain. *Journal of Experimental Biology*, 216(21):3963–3970, 2013.

[24] J. Wang, W. Qin, and L. Sun. Terrain adaptive walking of biped neuromuscular virtual human using deep reinforcement learning. *IEEE Access*, 7:92465–92475, 2019.

[25] C. J. Watkins and P. Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.

[26] Y. Zhu, Z. Wang, J. Merel, A. Rusu, T. Erez, S. Cabi, S. Tunyasuvunakool, J. Kramár, R. Hadsell, N. de Freitas, et al. Reinforcement and imitation learning for diverse visuomotor skills. *arXiv preprint arXiv:1802.09564*, 2018.