university of groningen

faculty of science and engineering

Deep reinforcement learning and reward shaping for a musculoskeletal model of a transfemoral amputee with a prosthesis during normal walking

Chandan Mundigehalla Sreedhara (S4250389)

August 19, 2022

Master Research Project

Dept. Of Artificial Intelligence

University of Groningen, The Netherlands

Internal supervisor(s):

Prof. Dr. Raffaella Carloni (Artificial Intelligence, University of Groningen)

Prof. Dr. Lambert Schomaker (Artificial Intelligence, University of Groningen))

Vishal Raveendranathan, Doctoral Candidate (Artificial Intelligence, University of Groningen))

Contents

Acknowledgments

I would like to thank Prof. Dr. Raffaella Carloni and Prof. Dr. Lambert Schomaker for the trust they put on me for doing this project as well as for their supervision and contribution.

I would also like to thank Vishal Raveendranathan for his more than useful input, help on the development of this project and for providing access to the prosthetic model.

Lastly, I would like to thank my family and my friends for the support given during the university years.

Abstract

Recent learning strategies such as reinforcement learning (RL) have favored the transition from applied artificial intelligence to general artificial intelligence. One of the current challenges of RL in healthcare relates to the development of a controller to teach a musculoskeletal model to perform dynamic movements. This project focuses on the design and training of deep reinforcement learning algorithm for the simulation of normal walking of the transfemoral prosthesis model. PPO with imitation learning along with reward shaping backed by biomechanical information is used to train the prosthesis agent on a flat terrain to walk with a constant speed. The agent is able to walk with a natural human gait after the training period. The results show that the emerging gait is similar to human gait based on the validation data. The gait is also evaluated with other metrics such as symmetry and metabolic costs which gives insight into the dynamics of prosthetic leg and the healthy leg. The forces generated by muscles and actuators in the transfemoral prosthesis is analyzed.

1 Introduction

Lower limb prosthesis can vastly improve the quality of life for amputees. Currently, development of prosthetic devices are influenced by long prototyping periods and increased costs of iterative design process. There has been a significant progress [1, 2, 3] in designs of motion controllers that are capable of adapting to the body's environment and achieve anatomically detailed and realistic simulation of human motion using accurate physics-based, biomechanical simulations. These simulations can greatly help in speeding up the development process of prosthetic devices by predicting the interactions between human amputees and prosthesis [4, 5]. Moreover, the simulations can be used to run several experiments with low cost to predict the viability of prosthesis in different environments [6, 7]. Many design approaches such as data-driven control [8, 9], stochastic optimizations [10, 11], reinforcement learning [12, 13], model-based optimization [14, 15] have been tested for robustness and control of prosthetics in the simulation. Recently, advances in deep reinforcement learning (DRL) have enabled major improvements in the realm of simulations of biomechanical models [16, 17, 18] which has accelerated the understanding of human-prosthesis interaction.

This project arises from the need to provide solutions to the population of transfemoral amputees, to accelerate the knowledge regarding the effect of amputation on the mechanical aspects of gait to the use of prosthesis. The simulation designed in this project primarily focuses on transfemoral amputee with a prosthetic model that consists of 15 muscles and 2 dynamic actuators using a DRL algorithm called Proximal Policy Approximation (PPO) [19] with imitation learning [20] in the simulation software of OpenSim [21]. Additionally, biomechanical information of the human gait and muscle excitations [22] is integrated into the algorithm in the form of reward shaping [23, 24] to guide the algorithm to converge faster to the desired gait. Finally, the project aims to analyse the muscle forces and actuator torques to validate the feasibility of the prosthesis model in real-world scenario.

1.1 Research Questions

This project aims to provide a concrete analysis and validation of a novel design of transfemoral prosthesis by simulating it in OpenSim framework using a DRL algorithm called PPO with imitation learning along with the novel idea of using biomechanical information in reward shaping. In contrary to the studies done by [22, 25] which uses biomechanical information of muscle excitations, this project proposes to use the biomechanical information as a part of reward function in the pure context of DRL to enable the convergence to the optimal policy of musculoskeletal model of 15 muscles and 2 actuators in an OpenSim software system. The validation of the prosthesis model is done by analysing the muscleforces and actuator torques to real world data after successfully running the simulation with human like gait. The gait is quantified by using the knee, ankle and hip angles, symmetry of the locomotion and muscle excitations which can further be used as a control input to the actual prosthesis model. This project hopes to pave the way for using deep reinforcement learning for the design of the control architecture and validation of transfemoral prosthesis models.

1.2 Scientific Relevance for Artificial Intelligence

This project is part of the Robotics Research Lab at University of Groningen which is involved in development of novel actuation systems and focuses on bio-inspired soft robots and lower-limb prosthetic device. The project aims to contribute to further expand the knowledge of the group by experimenting with the transfemoral prosthesis and validating the integrity of computer simulations in lower

limb prosthesis design. This project closely follows and builds upon the work done by Leanne de Vree [26] hence maintaining the continuity of research alongside of developing a novel approaches. As a bigger picture, this project contributes to the better understanding of human-prosthesis interaction which is key research area in robotics.

2 Literature review

OpenSim [21] is used to run all the experiments as it allows for array of utilities like modelling, controlling, simulations and analysis of human locomotion. Several studies [18, 27] have used OpenSim in simulations of human locomotion which proves its value and authenticity. It also runs inverse kinematics, controlled muscle control which are accurate tools to study gait patterns and evaluate datasets.

Previously, control models based on neuromechanics [28, 29, 22] and muscle dynamics [30, 31, 32, 32, 33] with the help of biomechanical information have been developed for human locomotion. Even though these models aid in simulation and understanding the human motions, it remains a challenge to adapt the models for dynamic environments and often struggle with high dimensional parameter spaces. Previous researches [34, 35] in Deep Reinforcement Learning (DRL) have shown their ability to learn control policies that operate directly on high dimensional, low- level representations of the underlying system. In the Artificial Intelligence for Prosthetics challenge [18] held by NeurIPS 2018, several participants submitted a working solution for transibital (below knee) amputee prosthesis which proves the effectiveness of DRL in lower limb prosthetic control. Recently, Leanne de Vree at Robotics Research Lab, University of Groningen used PPO with imitation learning [26] to successfully simulate the transfemoral model using a reduced muscle model and validated it with a public data set [36]. This project closely builds upon the work done by [26] and aims to bridge the gap between the real-world implication and simulations by using an actual prosthesis model made up of 15 muscles and 2 dynamic actuators designed in OpenSim by Vishal Ravindranthan of Robotics Research Lab, University of Groningen.

Furthermore, the idea of reward shaping using biomechanical information is used in the PPO with imitation learning for faster convergence. Reward shaping in Deep Reinforcement Learning algorithms have been effectively used in various domains [37, 38, 39] to transfer the expert knowledge of the respective domain to fasten the learning of the algorithm. Extensive studies are done in the effects of reward shaping on exploration [40], policy guidance [41] and transfer learning [42]. A theoretical framework of reward shaping and its application to accelerate convergence and learning is given by various authors [23, 24, 41] in previous years.

The usage of biomechanical knowledge to simulate human locomotion has been tried in different frameworks by various studies [22, 25, 28, 29]. The study done by [25] uses the biomechanical information of muscle activations given by electromyography (EMG) data to facilitate muscle coordination between 284 muscles through supervised learning as a part of bigger algorithmic process of DRL. Similarly, the study done by [22] uses biomechanical information exclusively in forming a control module of lower limb consisting of 22 muscle tendon units through algorithmic neural circuit.

The Table 1 gives a summary of different algorithms and control models used in DRL for human locomotion. The study proposed here uses the concept of reward shaping combined with biomechanical information of muscle excitations in the context of deep reinforcement learning to facilitate interaction between human and transfemoral prosthesis and validate the prosthesis model using the simulation framework given by OpenSim.

Article	Algorithm	Optimization	Input	Objective function	Output	Validation method	Simulation model
[43] 2017	Actor Critic DRL method	DNN with gradient descent	Electro-myography signals from the user	Minimize the errors in joint angles	Continuous action values	Ability to achieve desired joint angles	Powered robotic limb
[44] 2017	Trajectory mimicking with high and low level controllers	Gradient Descent	Reference data of pose and joint angles	Maximize reward by minimizing the loss	Joint angles, torques and muscle activations	Reference data and root mean square error	3D biped model with joint coordinates at knee, hip and ankle
[12] 2018	Proximal Policy Optimization (PPO) with imitation learning	DNN with gradient descent	Reference kinematic data	Maximize the reward for imitation learning by minimizing the loss	Muscle activations	Reference data and root mean square error (RMSE)	3D humanoid with joint coordinates
[45] 2018	Proximal Policy Optimization	DNN with gradient descent	Reference kinematic data	Minimize the error between simulation and the reference data	Target join angles and velocities	Reward from PPO	Bipedal robot
[25] 2019	Imitation learning and supervised method for learning muscle coordination	Two Deep Neural Networks (DNN) with gradient descent	Reference kinematic data and trajectory data from an experiment	Minimize the loss of imitation learning and muscle coordination	Muscle activations	Reference electromyography (EMG) data.	Full-body musculoskeletal model with 346 muscles

[46] 2019	PPO with imitation learning	DNN with gradient descent	Reference kinematic data	Maximize the reward for imitation learning by minimizing the loss	Muscle activations	Ground reaction forces, joint angles and muscle activations of reference data	Healthy human leg with 11 muscles in each adopted from [25]
[47] 2021	Direct Collocation	DNN with gradient descent	Actions and muscle states	Maximize custom reward functions based on biomechanical concepts	Joint angles, muscle activations and action outputs	Increase in rewards	Musculoskeletal model that contains 18 muscles (9 per leg)

Table 1: State of the Art of algorithms and control models used for Human Locomotion using DRL

3 Theoretical background

3.1 Deep Reinforcement Learning

Reinforcement learning (RL) algorithms involve the strategy of learning via interacting with the environment using sequence of actions, observations and rewards. A RL framework allows an agent to learn from trial and error. The RL agent recieves a reward by acting in the environment in such a way that the actions selected by the agent should maximize the expected cumulative reward over time. In other words, the agent, by observing the results of the actions taken in the environment, learns an optimal sequence of actions to execute in order to reach its goal. The general framework of reinforcement learning is depicted in the Figure 1.

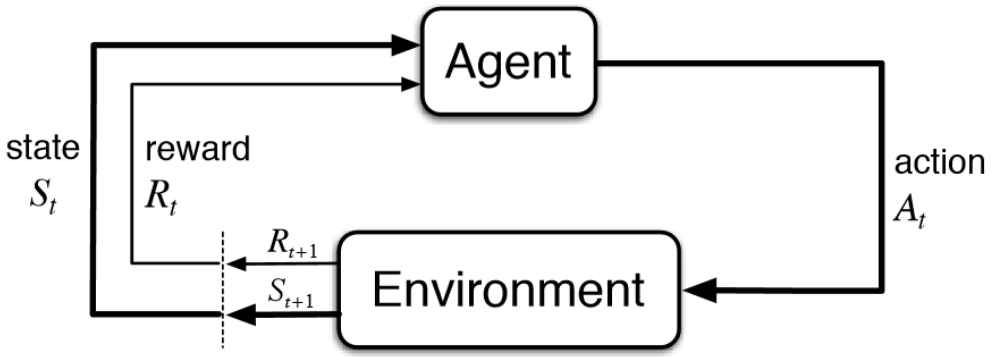


Figure 1: General environment of reinforcement learning.

Formally, a reinforcement learning agent is modelled as a finite Markov decision process (MDP). At each time step t, the agent select an action a_t from a set of allowed actions $A = \{1, ..., k\}$ at state $s_t \in S$, where S is the set of possible states. The action selection is done based on the policy on which the agent is being trained on. The policy, in essence, is a description of the behaviour of the agent's choice of action for each possible state and is denoted by π. As a result of each action, the agent recieves a reward $r_t \in R$, and observes next state $s_{t+1} \in S$. To put it in terms of probability, each possible next state s_{t+1} comes from a transistion distribution which is,

$$P(s_{t+1}|s_t, a_t), \quad s_{t+1}, s_t \in S, \quad a_t \in A(s_t) \tag{1}$$

Similarly, the probability of each possible reward r_t comes from a reward distribution

$$P(r_t|s_t, a_t), \quad s_t \in S, \quad a_t \in A(s_t) \tag{2}$$

Hence the expected reward recieved at time t, r_t, by executing an action a in current state s are calculated by,

$$E_{P(r_t|s_t,a_t)}(r_t|s_t = s, a_t = a) \tag{3}$$

The general aim of the agent is to learn the optimal policy π such that the sum of the discounted rewards over time is maximized. The expected discounted return R at time t is defined as follows:

$$R_t = E\{r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + ...\} = E\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k}\right] \tag{4}$$

where $E[.]$ is expectation with respect to the reward distribution and $0 \geq \gamma \geq 1$ is called the discount factor.
Considering transition probabilities and the expected discounted rewards, an action-value function $Q^{\pi}(s,a)$ can be defined as the expected return achieveable from state s, $s \in S$ and performing action a, $a \in A$ by following the policy π which is given by,

$$Q^{\pi}(s,a) = E_{\pi}[R_t|s_t = s, a_t = a] = E_{\pi}\sum_{k=0}^{\infty}\left[\gamma^k r_{t+k}|s_t = s, a_t = a\right] \tag{5}$$

A reinforcement learning agent aims to find an optimal policy, π^* which achieves the maximum future reward hence as a result, an optimal state-value function $Q^*(s,a)$. From dynamic programming, an iterative update for estimating the optimal state-value function is defined as follows:

$$Q_{i+1}(s,a) = E_{\pi}\left[r_t + \gamma max_{a'} Q_i\left(s',a'\right)|s,a\right] \tag{6}$$

where $s,s' \in S$ and $a,a' \in A$. The iteration converges to the optimal action-value function, Q^* as $i \Rightarrow \infty$ and is called a value iteration algorithm [48]

In real world applications, like in human locomotion, number of states and actions are very large and it becomes impossible to use table pairs of state-action values. A practical way to tackle this is to use the function approximator as an estimator of action-value function. The approximate value function is parameterized as $Q(s,a;\theta)$ with parameter vector θ. The parameters for the approximate value function can be learnt by minimizing the following loss function of mean-squared error in Q-values:

$$J(\theta) = E\left[\left(r + \gamma max_{a'} Q\left(s',a';\theta\right) - Q\left(s,a;\theta\right)\right)^2\right] \tag{7}$$

where $r + \gamma max_{a'} Q\left(s',a';\theta\right)$ is the target value. In terms of reward, the parameterized policy for the maximal expected reward can be written by changing the Equation 7 to encorporate reward and episodes as follows,

$$J(\theta) = E_{\pi}[r(\tau)] \tag{8}$$

These approximate value functions can be estimated with the use of neural networks which uses gradient-descent to minimize the error and obtain optimal policy. In the applocation of human locomotion, the approximate function is non linear and very complex which demands the use of deep neural network which has seen considrable success in approximating the high dimensional functions [49, 50, 51].

Deep learning techniques have the capability to extract features or representations from high dimensional data to learn multiple levels of abstractions. A typical example of deep nerural network architecture is given in the Figure 2. Neural networks are comprised of a node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network. The interconnectedness of the deep neural network makes it possible to learn, given enough data, non-linear functions, abstract patterns and high representations. This makes the deep neural networks a very powerful tool in solving the approximate value function. Hence, the deep neural networks are used as approximate value functions in reinforcement learning, where the weights between the nodes and thresholds are the parameters,

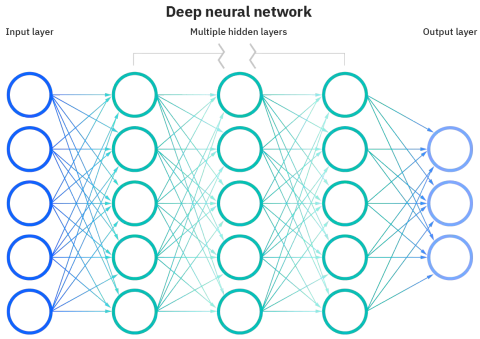


Figure 2: A general architecture of neural network.

leading to the field of deep reinforcement learning. By using Equation 8 and the policy gradient theorm [52], the gradient descent of the policy for deep neural networks can be written as,

$$\nabla E_{\pi_\theta}[r(\tau)] = E_{\pi_\theta}\left[\left(\sum_{t+1}^{T} G_t \nabla \log \pi_\theta(a_t|s_t)\right)\right] \tag{9}$$

where, τ is the trajectory of the episode and G_t is the total return at time t. However, learning the parameters of all the layers imposes a difficult optimization task which also comes with high computational complexity [53]. The next section discusses the optimization method that is used by this project.

3.2 Proximal Policy Optimization (PPO)

The optimization used in this project is called Proximal Policy Optimization (PPO) which is a policy gradient (PG) method that can be used for both discreet and continuos action spaces. Firstly, it collects a set of trajectories for each epoch by sampling from the latest version of the stochastic policy. Then, the rewards and the advantage estimates are computed in order to update the policy and fit the value function. The policy is updated via a stochastic gradient ascent optimizer, while the value function is fitted via gradient descent algorithm. This procedure is applied for many epochs until the environment is solved. The objective function of the PPO is given as,

$$L^{clip}(\theta) = \hat{E}_t[min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t)] \tag{10}$$

where,

θ is the policy parameter

$\hat{E}_t$ denotes the empirical expectations over timestamp t

r_t is the ratio of the probability under the new and old policies

$\hat{A}_t$ is the estimated advantage at time t, here the advantage is the difference between an expected and real reward from action a

ε is a clipping hyperparameter

From the Equation 10, PPO computes an expectation over a minimum of two terms: normal PG objective and clipped PG objective. The key component comes from the second term where a normal PG objective is truncated with a clipping operation between $1-\varepsilon$ and $1+\varepsilon$. The updated policy will be ε-clipped to a small region so as to not allow huge updates which might potentially be irrecoverably harmful. In short, it also ensures that the old policy and new policy are at least at a certain proximity (denoted by ε), and very large updates are not allowed. The next sections give an overall of the rewards that are used in the reinforcement learning.

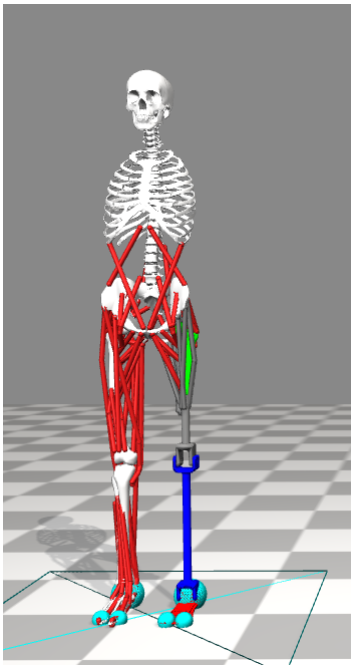


Figure 3: Original prosthetic model designed by [55] which contains 76 muscles and 2 actuators.

4 Materials

This section gives the description of the experimental dataset used for the training of deep reinforcement learning and the prosthesis model that is used as an agent in this project.

4.1 Dataset

The dataset [54] used in this project is a curated experimental data of human walking locomotion which contains the joint angles and velocities. The dataset was also given by the NIPS competition [18] where the participants used the dataset to evaluate their models. Furthermore, the dataset is validated by running computed muscle control (CMC) and inverse kinematics on OpenSim software and it is confirmed that the coordinates and orientations of the dataset is in sync with the reinforment learning environment of the prosthesis model. The dataset contains the walking locomotion of human for 10 steps from heel to heel strike with the average walking velocity of 1.25 m/s. The joint angles and velocities are sampled at 100 Hz for total of 13 seconds which sums upto 16.25 meteres in distance. It should also be noted that the dataset required no scaling or preprocessing since it is already prepared for OpenSim environment.

4.2 Prosthetic model

The prosthetic model used is a reduced model of a forward dynamics computed model done by the doctoral candidate Vishal Raveendranathan [55]. The original model is shown in the Figure 3. The original model from [55] consists of two actuators and 76 muscles where as the reduced model contains lesser muscles in order to facilitate reinforcement learning without parameter explosion. The comparision between the original and reduced model is shown in Figure 4. In particular, the reduced model has 15 muscles and 2 actuators. The 2 actuators constitute the transfemoral prosthesis where

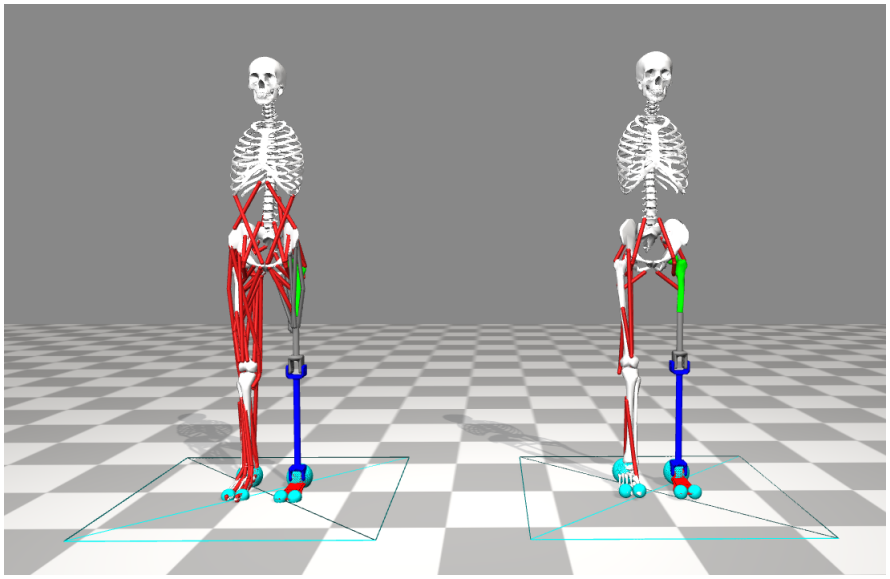


Figure 4: Original prosthetic model designed by [55] in comparision with the reduced muscle model which contains 15 muscles and 2 actuators.

each actuator is situated at the knee and ankle joints. The Table 2 gives the details of the muscles and actuators that constitutes each leg.

Left leg	**Right leg**
Hip Abductor	Hip Abductor
Hip Adductor	Hip Adductor
Hip Flexor	Hip Flexor
Gluteus Maximus	Gluteus Maximus
Knee actuator	Hamstrings
Ankle actuator	Biceps Femoris
-	Rectus Femoris
-	Vastus Medialis
-	Gastrocnemus
-	Soleus
-	Tibialis Anterior

Table 2: Details of the muscles and actuators of the reduced musculoskeletal model

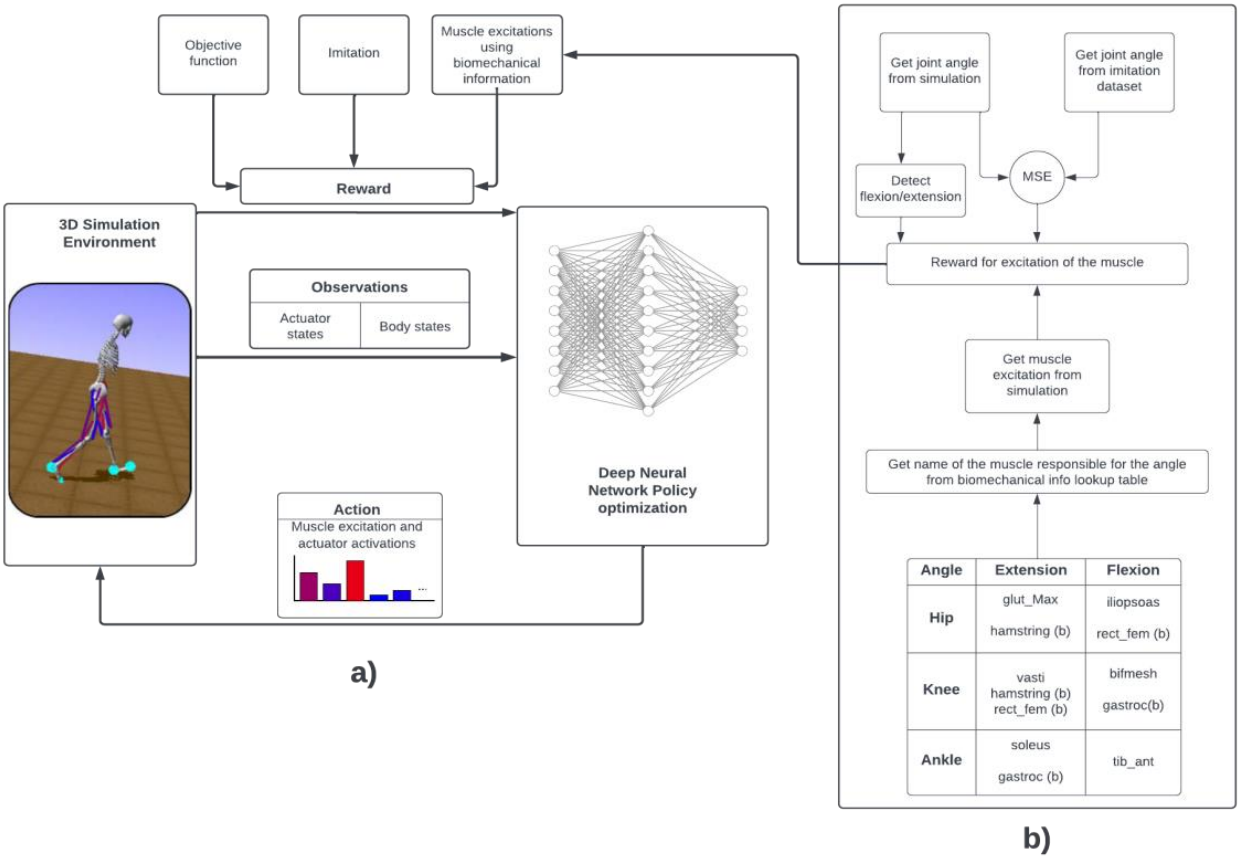


Figure 5: DRL with imitation learning and reward shaping using biomechanical information of muscle excitation. **a)** Simulation environment with DRL framework. Environment takes an action as input, musculoskeletal model for one-step and provides reward and observation. **b)** Reward shaping block diagram based on muscle excitation.

5 Methods

In this section, the overall system architecture, algorithms used and the entirety of framework are explained in detail.

5.1 System architecture

This project proposes an architecture depicted in the Figure 5. The architecture is extended off the deep reinforcement learning to add imitation learning and the reward shaping. The architecture is composed of single neural network, which is used to optimize the policy for maximum rewards using PPO, a reward function to facilitate the rudimentary locomotion action, imitation learning to follow the gait based on the experimental data and biomechanics based reward shaping to reinforce the coordination between the healthy and prosthetic leg.

5.2 Neural network

The neural network used in the project consists of an input layer of the size 94, hidden layer of size 200 and output layer of size 17. The input layer corresponds to the observation space of the

musculoskeletal model which consists of joint positions, joint velocities, muscle activations, muscle forces and actuator states. The number of hidden units is based off the heuristic experiments and the complexity of converting the observation space to action space. The output layer has a size of 17, where first 15 unit corresponds to the activation of muscles and the last 2 units corresponds to the activaton of knee and ankle actuators respectively.

5.3 Reward function

Reward is a incentive mechanism that gives the agent a numerical score based on the action taken. In essence, a reward function is a mapping of each percieved state of the environment to a single number, specifying the intrinsic desirability of that state. In this project, since the agent desires to walk, the general reward function deals with the action of moving forward in a constant velocity. The overall goal of this reward function is to generate controls such that the agent moves forward. This is a rudimentary reward aimed at generating the locomotion without a concern of gait. Mathematically, this reward function is defined as,

$$penalty_x = |v_x(s_t) - d_{t,x}|^2$$

$$penalty_z = |v_z(s_t) - d_{t,z}|^2$$

$$reward = e^{-8(penalty_x + penalty_z)} \tag{11}$$

where,

$v_x(s_t)$ and $v_y(s_t)$ are velocity of the agent in x and z coordinates respectively

$d_{t,x}$ and $d_{t,z}$ are desired constant velocities for the agent in x and z coordinates respectively

This reward the agent for adhering to a desired constant velocity by issuing penalties whenever there is a deviation from it. The goal of this reward is to habituate the agent to develop a walking pattern in the specific target velocity. During each time-step, the difference between the velocity of the agentâĂŹs pelvis and the target velocity is calculated in x and z coordinates. The y coordinate is not considered since the walking pattern does not involve any change in motion in the y-axis such as jump or crouch. The sum of squared error of the differences in x and z denotes a penalty for deviating from the target velocity. Then the reward is calculated by adding those penalties with a negative exponential. The reason to add the exponential is three fold: One, it scales the reward between 0 to 1. Two, it allows to control the strictness of the reward granted to the agent ie, higher the exponential, lower the tolerance towards the devaition of the agent and hence lower the reward. Third, the exponential is also a well suited function for calculation of deep neural network gradients.

5.4 Imitation learning

The reward function ensures that the agent moves forward in a fixed velocity but it doesn't ensure the human like locomotion. Hence, the imitation reward is added to the given general reward function. Imitation learning has proven [35, 20, 26] to increase the performance and attain the natural gait of human walking. The imitation learning uses the curated experimental data to ensure the optimal

covergence of PPO to natural walking pattern. For each time-step t, the position and velocity loss of the pelvis, hip, knee and ankle joints are calculated as follows,

$$imitation\_loss_{pos} = \sum_j |p_j(s_t) - d(p)_{j,t}|^2$$

$$imitation\_reward_{pos} = e^{-2(imitation\_loss_{pos})}$$

$$imitation\_loss_{vel} = \sum_j |v_j(s_t) - d(v)_{j,t}|^2$$

$$imitation\_reward_{vel} = e^{-0.1(imitation\_loss_{vel})}$$

$$total\_imitation\_reward = (imitation\_reward_{vel} + imitation\_reward_{pos})/2$$

where,

$j \in$ pelvis, hip, knee, ankle

$p_j(s_t)$ and $v_j(s_t)$ are position and velocities of joint j

$d(p)_{j,t}$ and $d(v)_{j,t}$ are desired position and velocities of joint from experimental dataset

The losses computed in the joints serve as a penalty to the reward. The higher the losses, the lower the reward and the other way around. This encourages the agent to keep its states as close to the ones in the data as possible, hence encouraging a more natural walking pattern. The position reward is given more weightage for stricter adherence to dataset compared to the velocity reward since the general goal term in Equation 11 already penalizes the velocity of the entire agent's body. The total imitation reward given by subsection 5.4 is a sum of imitation reward of position and velocity which is then scaled between 0 and 1.

5.5 Reward shaping

The prothetic is made of actuators which have the classical mechanics properties whereas the rest of the legs is made of muscles which exerts force and controls the locomotion in a vastly different way compared to the actuators. The emergence of coordination between the set of actuators and muscles is necessary to facilitate the human like locomotion. From the perspective of deep neural networks, the coordination of two seperate sets of forces consists of having a higher dimensional distribution which represents the dynamics between the two. Even though this is feasible with DNNs and the convergence is possible without any reward shaping, it is unclear if it will certainly converge. Previous works [22, 25, 28, 29] have shown that provinding extra information of muscle cooridination and excitations have resulted in successful convergance of human-like gait. This project proposes to use the biomechanical information of muscle excitation during the gait phase to converge the neural networks to human like gait and as a result, learn the dynamics between the actuators of prosthetic leg and the healthy muscles.
The reward shaping uses experimental dataset [54] used by imitation learning but the motivation to use the dataset is to compute the reward for the excitation of the muscles rather than the imitation itself. Here, the experimental dataset [54] is used as a reference angle for a given joint and is compared

with the joint angle in the simulation using the mean squared error. Based on the reference angle, the movement (flexion/extension) of the joint is detected. Meanwhile, the excitations of the muscles responsible for the joint angle and the movement is taken from the simulation environment states using the lookup table. Based on the meansquared error and the muscle excitations, the reward is given to the agent. The mathematical form of the reward at time t is shown below,

$$mse_j = |a_j(s_t) - d_{t,j}|^2$$

$$reward = \sum_j \sum_m e^{-(mse_j \times exc_m)} \tag{12}$$

where,

$j \in \{hip, knee, joint\}$

$a_j(s_t)$ is the angle of joint j of the agent

$d_{t,j}$ is the angle of joint j from expereimental dataset

mse_j is the mean squared error of the joint j

exc_m is the excitation of the muscle m, where $m \in$ set of muscles M taken from the lookup table

The reward is then scaled between 0 and 1. The core idea behind this reward is to make the explicit connection between the excitation of the muscles and the joint angles explicit. Hence, the final reward given in Equation 12 contains both the terms of excitation of the muscle and joint angles being multiplied. The multiplication of these terms is a way to indicate that they reinforce each other and consequently, converge to a successful coordination between the healthy leg and the prosthetic leg. The reward shaping is also served as an extension to the general deep reinforcement learning framework to have a control over the learning process by attaching weights to the muscle excitations.

5.6 Performance Criteria

Performance of reinforcement learning is monitored using these criterias,

Cumulative Reward: The expected trend is that reward should consistently increase over time even though Small ups and downs are to be expected as a result of explorationa and complexity of the task. The cumulative reward is constantly monitored in order to gauge the performance of the agent in the training process.

Value Loss: A good indicator for the performance is the value loss since values will increase as the reward increases, and should decrease when reward becomes stable. That is, when the agent's policy is stable enough, the value loss should decrease.

5.7 Evaluation methods

This section gives the methods that are chosen to evaluate the model's gait.

5.7.1 Root mean squared error (RMSE) of angles

The experimental data [54] is used to comapare the joint angles and positions with the trained model. Specifically, joint angles of hip, knee and ankle is compared for all the gait cycles. For every joint, RMSE is reported as a quantitative evaluation. The formula for RMSE of a joint j is given below,

$$RMSE = \sqrt{\frac{\sum_{i=0}^{N}(j(i) - \hat{j}(i))^2}{N}} \tag{13}$$

where,

N is the total number of timesteps in the total gait

$j(i)$ and $\hat{j}(i)$ is actual joint angle from dataset and joint angle from the simulation respectively.

5.7.2 Symmetry angle

A human locomotion gait is a result of interaction between multiple body segments and and effect of complex dynamics. Replacing a limb with the prosthesis which has actuators rather than muscle is a major change in this complex dynamics and the deviation in the gait due to this change is a common phenomenon [56]. This phenomenon is usually accompanied with change in symmetry of the locomotion. An average able bodied individual has very minor asymmetries in the gait whereas an individual with prosthetics is bound to have an asymmetric gait [57]. Hence, a viable way to measure the quality of the locomotion with prosthetics is to measure the symmetry of the gait itself. Previous studies [58, 59] have shown various measures to quantify the gait symmetry and this project uses the measure of Symmetry Angle (SA) [60] due to its robustness and a standard scale for interpreting the measured asymmetry. The formula for symmetry angle is given as follows,

$$SA = (45^\circ - arctan(X_a/X_u))/90^\circ * 100\% \tag{14}$$

where,

X_a is the step length of prosthetic leg

X_u is the step length of healthy leg

An SA value of 0% indicates perfect symmetry (equivalence between the values), and a value of 100% reflects two values that are equal and opposite in magnitude.

5.7.3 Metabolic Cost

The metabolic cost of walking is the energy expended by the human body to move a certain distance. The metabolic cost can be calculated using variables that are studied in gait analysis, such as joint moments, joint power, or muscle forces, lengths and activations [61]. Studies [62, 63, 64] have been done to measure the metabolic cost of walking in an average human. Hence, the effect of intervention of prosthetics can be evaluated by comparing the metabolic costs of the agent with prosthetics to the cost of the locomotion of average human. There are number of metabolic models [65] that give the cost based on the variables that are provided. This project uses the model proposed by [66] due to its accessibility to variable choices and robustness to calculate the metabolic cost. The formula to calculate the cost is given as follows,

$$C_{calc} = \frac{1}{Tmv} \int_{t=0}^{T} \sum_{i=1}^{N_{mus}} \dot{E}_i dt \tag{15}$$

where,

T is duration of motion

m is participant's mass

v is speed

N_{mus} is number of muscles

$\dot{E}_i$ is energy rate of the muscle

6 Experimental Setup

This section gives the practical details about the experimental setups, technologies used and hyperparameters.

6.1 Tools and technologies

OpenSim is used as an interface between the prosthetic model and the reinforcement learning algorithm. OpenSim is a well-established human locomotion simulator. It allows to perform inverse kinematics, controlled muscle compute and forward dynamics of musculoskeletal models. Furthermore, it provides a flexible platform to run reinforcement algorithms which make use of enormous amount of simulations in combination with optimization routines. This project adapts OpenSim as its simulation environment and an interface to the control module of the trained model. In particular, OpenSim provides the interface to get the observations of the joint positions, angles and accelerations along with the muscle and actuator states. These observations are used to train the reinforcement algorithm to learn the dynamics of the prosthetic model. Several experiments of reinforcement algorithm are conducted using the OpenSim environment to get the optimal results.

All the experiments and simulations are done in a dedicated desktop provided by Robotics Research Lab, University of Groningen whose specifications included AMD Ryzen 7 3700X 8-Core Processor with 32 GB of memory.

The implementation of Proximal Policy Optimization (PPO) from OpenAI's Stable Baselines [67] is used along with OpenSim's simulation environment. The entire training routine and experimentation is written in Python3 [68] and the deep neural network uses TensorFlow [69] library of Python3.

Parallel processing is used to train multiple workers of the PPO algorithm to boost the performance and reduce the training time. The Python library MPI4Py (Message Passing Interface for Python) [70] is used as an architectural base for implementation of multiprocessing.

6.2 Hyperparameters

The main hyperparameters that were considered during the simultions are given below:
PPO gathers trajectories (a series of state, action, rewards) as far out as the horizon limits, then performs a stochastic gradient descent update of minibatch size on all the gathered trajectories for the specified number of epochs. The hyperparameters to consider here are,

Horizon: This corresponds to how many steps of experience to collect per-agent before adding it to the experience buffer. This number should be large enough [71] to capture all the important behavior within a sequence of an agent's actions. Taking into consideration of the available memory, the episode lengths and the paralell workers, this is set to 1536.

Minibatch Size: Corresponds to how many experiences are used for each gradient descent update. This should always be a fraction of the Horizon Range. The minibacth range is set to 512.

Epochs: This is the number of passes through the experience buffer during gradient descent. The larger the minibatch range, the larger it is acceptable to make this. However, decreasing this will

ensure more stable updates, at the cost of slower learning. Typical range usually includes 1-10 depending on the computational resource available to the user. Considering the availability of parallel processing and raw computing power, the epoch range is set to 4.

Step size: Corresponds to the strength of each gradient descent update step. This should typically be decreased if training is unstable, and the reward does not consistently increase. After experimentation, this is set to 1e-3.

If the policy is updated in too large a step, policy performance can collapse drastically and never recover. PPO uses a surrogate loss function as given in Equation 10 to keep the step from the old policy to the new policy within a safe range. The hyperparameters considered in this regards are,

Clipping Parameter: Corresponds to the acceptable threshold of divergence between the old and new policies during gradient descent updating. Setting this value small will result in more stable updates, but will also slow the training process. The value is set to 0.2 so that it is stable yet also allows for slightly bigger update of the policy.

Entropy Coefficient: Corresponds to the strength of the entropy regularization, which controls the randomness of the policy. This ensures that the agents properly explore during training. Increasing this will ensure more random actions are taken hence increasing the exploration. Typical value ranges from 0.1 - 0.0001. In order to have a balance between the exploration and exploitation of the agent, this value is set to 0.01.

Gamma: Corresponds to discount factor which determines how much the agent cares about rewards in the distant future relative to those in the immediate future. If it is set to zero, the agent will be completely myopic and only learn about actions that produce an immediate reward. This is set to 0.999.

Lambda: is a smoothing parameter used for reducing the variance in training which makes it more stable. This is set to 0.9.

In addition to PPO parameters, the neural network used has two hidden layers with each layer consisting of 200 neurons and uses *tanh* as activation function.

7 Results and discussion

In this section, the results that are obtained from the simulation are given.

7.1 Training rewards:

The training reward is a definitive indication of the performance of algorithm. As seen in the Figure 6, the reward increases linearly until 1200 iterations and then stabilizes until the end of iterations. The stabilization is an indication that the agent has learnt the policy, or in this case, a dynamics of coordination between the healthy and prosthetic leg such that it satisfies the objective function which is essentially a function of locomotion with human like gait.

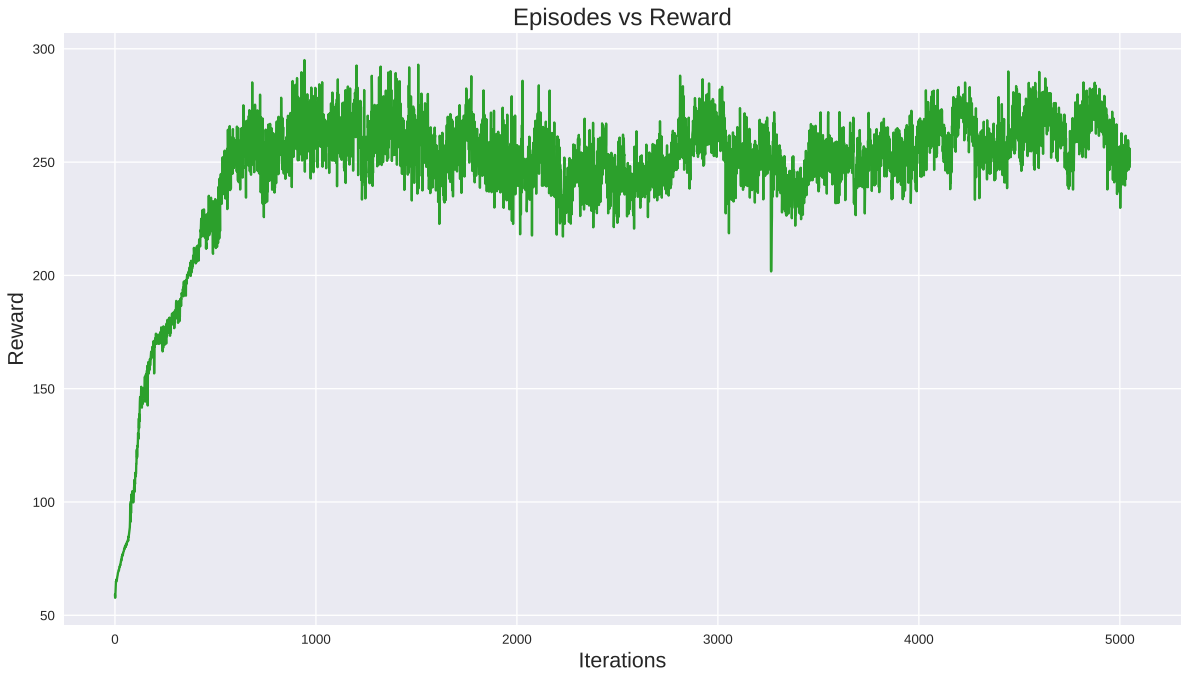


Figure 6: Average reward obtained by the agent in every iteration. The rewards increase linearly till 800 iterations and stabilizes to improve the policy.

7.2 Kinematics evaluation:

The kinematics of the agent is evaluated against the experimental dataset [54] and the comparative results for the healthy leg and the prosthetic leg for hips, knee and ankle are shown in Figure 7 and Figure 8 respectively.

The figures show that emerging kinematic data patterns are repetitive and similar to the experimental dataset except for the knee flexion in both healthy and prosthetic leg. The RMSE score between the kinematics of the simulation and experimental dataset is calculated to measure the error which is shown in the table Table 3 which also reflects the fact that both the knee has comparatively high error due to the lack of knee flexion.

	Prosthetic leg (left)	**Healthy leg (left)**
Hip	0.1916 (11.86°)	0.2702 (15.89°)
Knee	0.334 (19.16°)	0.3162 (17.94°)
Ankle	0.1521 (8.71°)	0.4713 (27.18°)

Table 3: RMSE between the kinematics of the simulation and experimental dataset given in radians and degrees inside the bracket

	Knee Actuator	**Ankle Actuator**
Mean Torques (Nm)	68.51	24.41

Table 4: Mean Torques of knee and ankle actuators

7.3 Muscle and actuator forces:

Muscle forces are calculated throughout the entire gait and the mean muscle forces are shown in the Figure 9. Since the actuators have rotational velocity, the mean actuator torques are given in the Table 4. In comparison, the four muscles in prosthetic legs collectively has more force than its counterpart in healthy leg. This is because the four legs are collectively trying to perform the tasks of the missing muscles along with the mechanical actuators.

The mean torque of the knee actuator is also taking a higher load compared to the ankle actuators. This might be due to the placement of the actuator, which is directly beneath the remaining muscles, where the muscle forces are being directly exerted into the knee actuator. This compels the knee actuator to consistently have a high torque.

The normalized torques and stiffness resulted from the knee and ankle actuators are plotted to get an understanding of the behavior of actuators compared to the classical muscle behaviors in healthy leg. Since the output forces of dynamic actuators consisted of a lot of fluctuations compared to the smoother behavior of the muscle forces, a simple moving average of window size of 9 and Savitzky-Golay filter is used to smoothen the output forces before plotting. The simple moving average gives a constantly updated average by calculating the average in the given window. The Savitzky-Golay filter is applied by using convolution on the data and by fitting successive subsets of adjacent data points with a low-degree polynomial. The torque and stiffness plots of the knee actuator and ankle actuator is given in the Figure 10 and Figure 11 respectively. The stiffness graph is obtained by plotting the actuator torque against the angle of the joint.

7.4 Symmetry Angle

Symmetry angle is measured for the step length of the trained agent, that is, it quantifies the gait deviation of the agent based on the heel to heel strikes. The point of measurement is the coordinates of right and left feet. In order to get an accurate estimation, the agent is simulated to walk the 10 gait cycles and the SA is calculated for each gait cycle. This simulation is repeated 10 times to get the mean of the SA for each gait cycle. The result for the 10 gait cycle is shown in the table Table 5,
It is seen that the first gait has a very high asymmetry which is due to the fact that the agent takes a big

Gait cycle	1	2	3	4	5	6	7	8	9	10
Mean SA %	40.12	14.24	12.79	12.45	12.69	13.12	13.03	12.95	12.90	13.38

Table 5: Mean SA of 10 simulations for each gait cycle

Joint	Hip	Knee	Ankle
Cost J/Kg/m	1.07	1.34	2.96

Table 6: Metabolic costs of the joints

step trying to stabilize itself effectively. As a result, the SA is high in the first gait cycle but decreases drastically in the consequent gait cycles.

7.5 Metabolic Cost

Metabolic costs is calculated for each joints of the healthy leg using the Equation 15. The Table 6 shows the metabolic costs of the joint of healthy leg. The metabolic costs are highest at the ankle and decreases considerably at knee and hips. Since the ankles are responsible for major weight lifting of the locomotion, the high cost equates to it. When compared with the literature [65] of metabolic costs of average human walking at 1.25m/s as the agent does, it is found that the cost is lesser at all three joints.

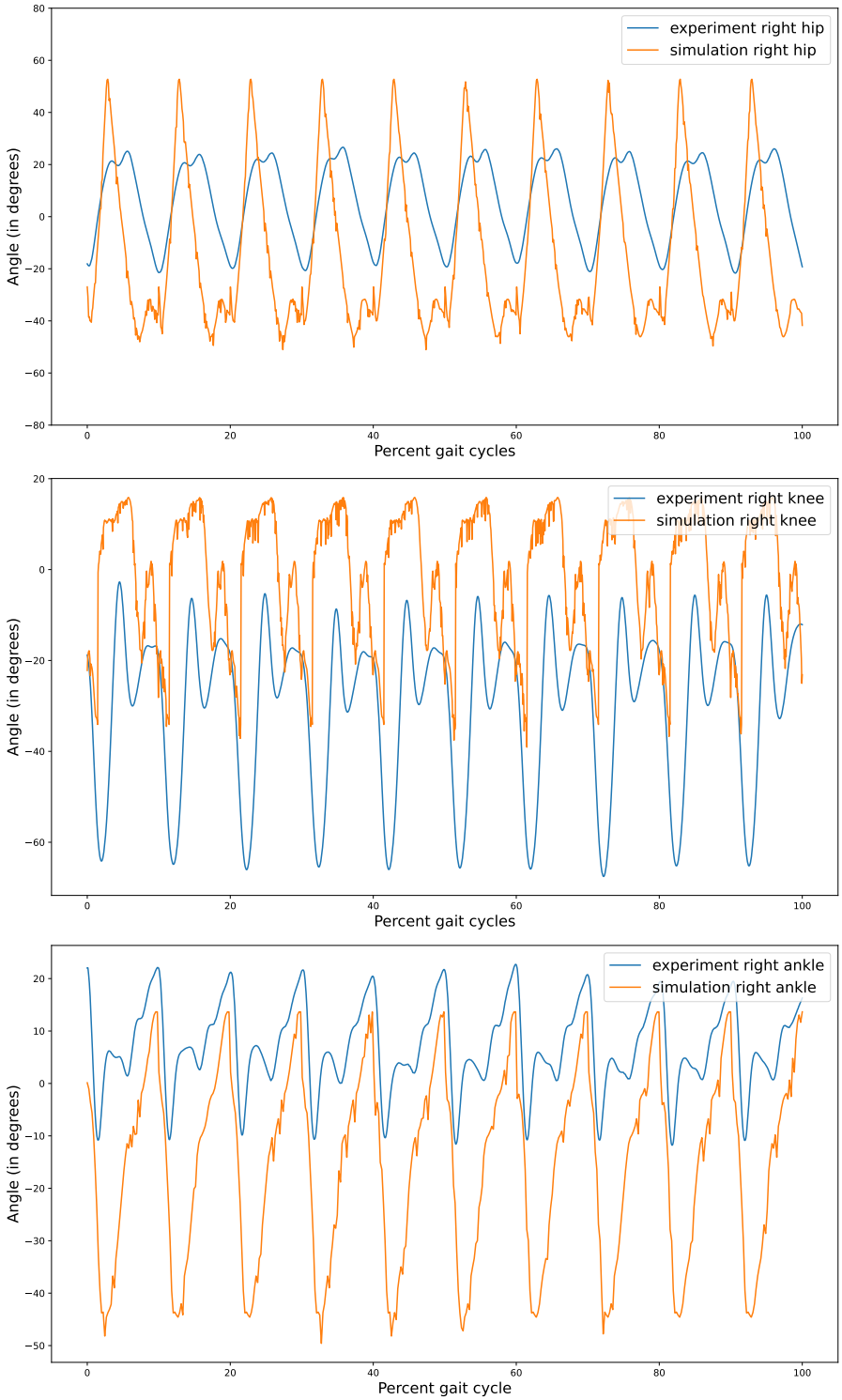


Figure 7: The emerging gait pattern of the simulated (orange) healthy right leg against the experimental dataset (blue). The figure shows the gait pattern comparision of healthy right hip (top), healthy right knee (middle) and healthy right ankle (bottom).

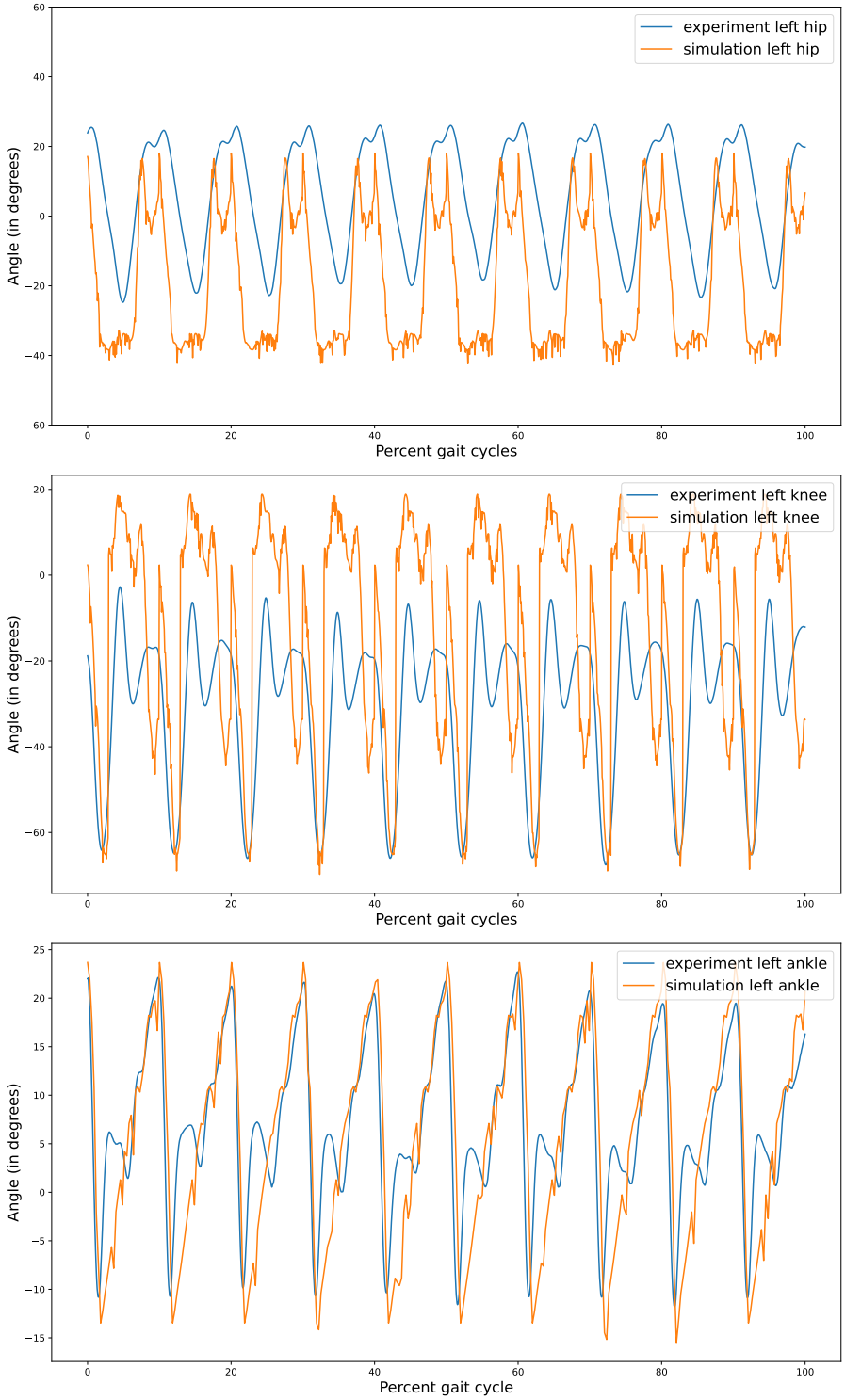


Figure 8: The emerging gait pattern of the simulated (orange) prosthetic left leg against the experimental dataset (blue). The figure shows the gait pattern comparision of left hip made of muscles (top), left prosthetic knee actuator (middle) and left prosthetic ankle actuator (bottom).

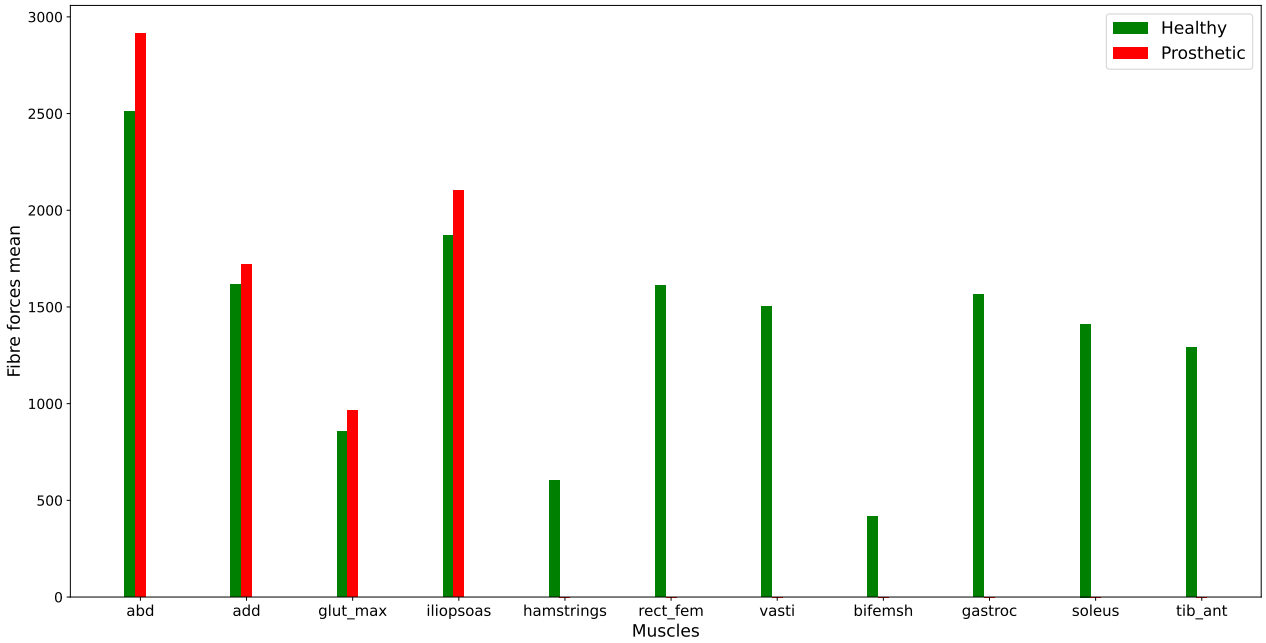
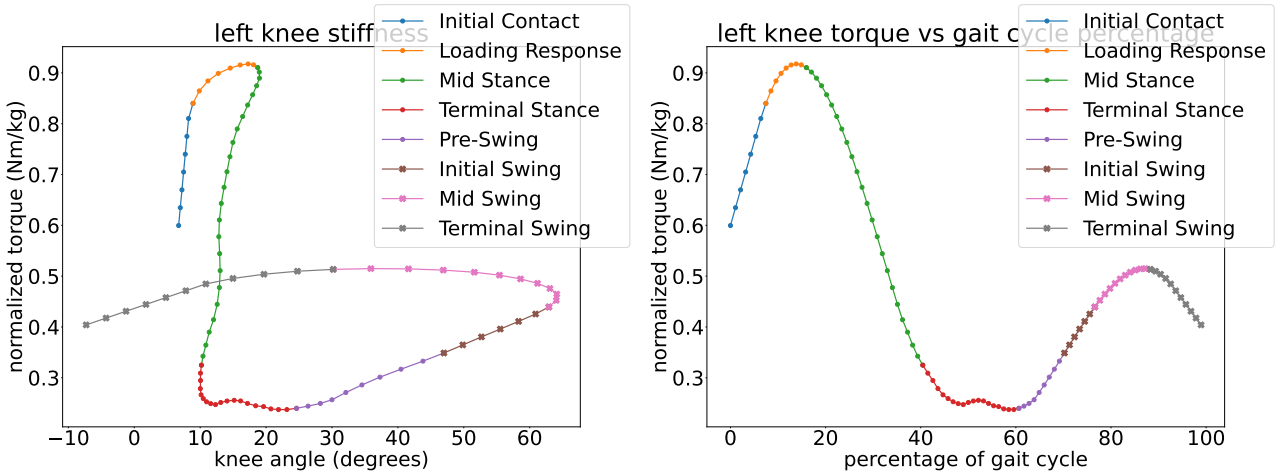
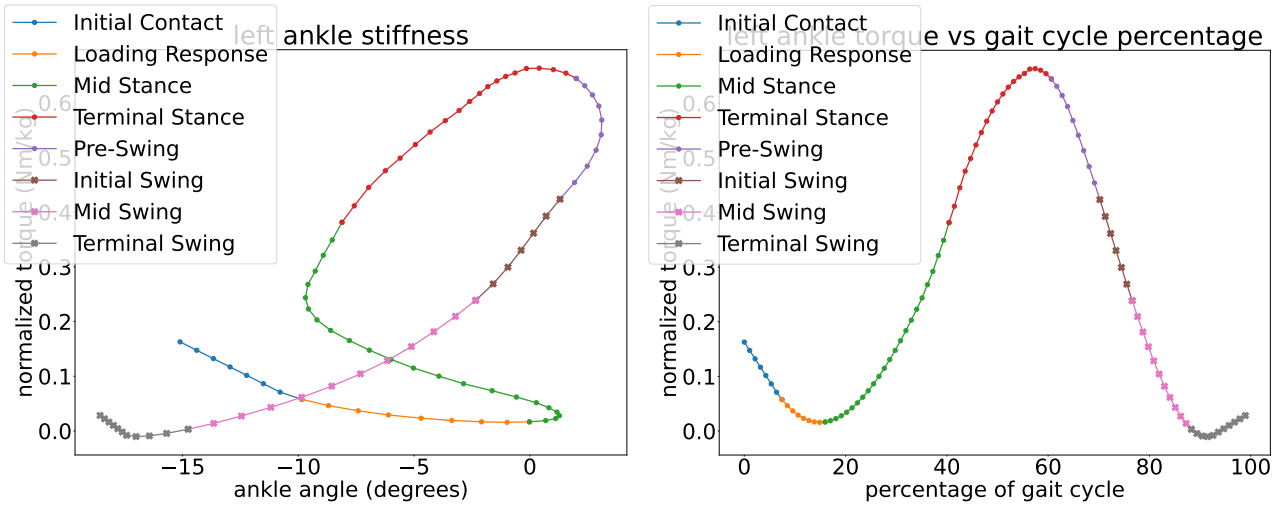


Figure 9: Mean muscle forces of healthy and prosthetic leg. Muscles in the prosthetic leg is found to have higher mean force compared to the respective muscles in the healthy leg. All forces are in Newtons (N)



(a) (b)

Figure 10: (a) Knee actuator stiffness plotted for one gait cycle using knee actuator torque against the angle of the joint. (b) Normalized knee actuator torque for one gait cycle



(a) (b)

Figure 11: (a) Ankle actuator stiffness plotted for one gait cycle using ankle actuator torque against the angle of the joint. (b) Normalized ankle actuator torque throughout one gait cycle

8 Discussion

Human locomotion control is a complex biological process developed by human beings from the first years of life. The addition of mechanical prosthetic to the natural biological process is a complicated endeavor in terms of feasibility, flexibility and adaptability. This project aimed to explore the validity and analysis of a novel design of such prosthesis using deep reinforcement learning simulations. The deep learning simulation approach faces many modeling assumptions related to the optimization objective function, lack of transparency and experimental uncertainties. The addition of bio-mechanical information as a reward to the objective function in this project is aimed at circumventing these modeling assumptions and integrating the domain knowledge to the reinforcement learning. The obtained outcomes were analyzed and evaluated by chosen metrics from both the bio-mechanical and reinforcement learning perspective. The deep reinforcement learning uses PPO as an optimization algorithm and converges to the locomotion gait with the normal walking speed. The reinforcement learning training rewards increases linearly for first 1000 iterations where the model rapidly learns the dynamics of the environment and its movement to obtain an optimal baseline policy. Then it steadily improves the policy to reach the optimal gait whilst having a steady increase in the rewards. The obtained gait closely follows the imitation data but there is a lack of knee flexion in the healthy leg which leads to a asymmetrical gait pattern as shown by the symmetry angle reported in subsection 7.4. The average muscle and actuator forces are within the range of realistic implementation given that the limit of these forces are higher than the human performance. The interesting aspect of the simulation is the behaviour of actuators since no restrictions are imposed on its behavior and they are dynamic in nature. Even though the prosthetic model moves in a steady speed, the observed force of the actuators that thrust the model forward is fluctuating in nature. Moreover, the average of these fluctuating forces gives an optimal force value which means the neural network in deep reinforcement learning as a controller of the actuators isn't implicitly learning to have a steady force but rather opting to give bursts of forces at a time. After smoothening the fluctuations using the filtering techniques, the plots of stiffness and torques display the curves similar [72, 73] to the knee and ankle joints composed of muscles. This similarity in curves is a evidence to the fact that the deep reinforcement learning is capable of learning the control values to the actuators.

The Figure 7 and Figure 8 show the angles followed by the joints of healthy and prosthetic leg compared to the experimental data. In the healthy leg, the knee follows the same pattern as experimental data but diverges in the angles. The lack of flexion in healthy knee is clearly evident here and the knee overextends as well. This lack of flexion and the overextension can be the result of overfitting of muscles around the knee for extension and hence resulting in low flexion. Similar phenomenon is seen in healthy ankle as well. In contrast, the actuator knee and ankles fit realtively well compared to the experimental data which supports the assumption that the reinforcement learning algorithm finds it harder to coordinate between the muscles and relatively easy to tune the actuators. This assumption is based on the fact that the range of control values for the actuator is less compared to that of range of values in coordination of activations of the muscles involved in the flexion and extension.Another peculiar pattern to notice in the Figure 7 and Figure 8 are the sharp edges and irregularites in the curves of simulation compared to experiment. This points at the inability of reinforcement learning policy to capture the continuity of the muscle activations and actuator torques in a smoother fashion. Given the fact that no explicit reward was given for smoothness, the policy is only concerned with reducing the loss between the imitation data and simulation. It might have been the case that it chose to fit the overall pattern rather than the details of smoothness to reduce the overall loss.

The stiffness and torque plots given in Figure 10 and Figure 11 shows the post processed filtered torques of the knee and ankle actuator. The fluctuations in the torques are due to the lack of smoothness controling mechanism in the algorithm and these fluctuations in the knee and ankle of prosthesis might also be the cause of fluctuation and sharp curves in healthy leg motion. Since the locomotion is a result of coordination between the prosthetic leg and the healthy leg, the policy might choose the irregularities seen in the actuators as a natural motion and imitate the same in healthy leg. The knee actuator takes a longer time at the terminal stance and the torque gradually drops at the terminal swing phase. The nature of torques is that the knee actuator exhibits lowest torque while the ankle actuator exhibits its highest torque during the terminal stance phase indicating the shift of forces from knee joint to ankle joint. Conversely, during the initial contact, the knee actuator exerts more force while the ankle actuator is at its low. The torque plots of both knee and ankle forms a closed ellipsoid which can be interpreted as the prosthetic model performing a full cycle of gait.

The use of biomechanical mechanical information in reward shaping to activate the muscles is mainly used to control the learning process of the deep reinforcement learning to converge to the expected gait. The weight of the reward given to each of the muscle joints is reflected at the obtained gait. The tuning of the weights is a major part of the training and the underlying consequences of adjusting weights is not fully explicit due to the complexity of the ebvironment and the model. However, it is seen in the experiments that the reward shaping using the domain knowledge of biomechanics to influence the activations of muscles is a useful tool to control the learning process. Furthermore, the model obtained can be used in real-time due to the small size of the model to test the validity of an actual prosthesis by inferring to the control values and running on-line cloud learning.

The use of deep reinforcement learning for locomotion in context of prothesis is successful in obtaining the policy that is capable of imitating the data roughly. However, it is not able to capture the distilled nuances of the biological process of locomotion in terms of smoothness, reduced torques and quality of the natural human walking. The policies obtained using reward shaping method is able to influence the muscle activations and combined with the imitation reward, it gives an estimation for validation of the prosthesis. The fluctuations and erraticness is the result of lack of direction in terms of rewards and penalty for the deep reinforcement learning to completely imitate the natural human walking. Given the fact that no explicit equations or dynamic control formulas are used in the algorithm and yet it is able to get a comparable gait is a good indication for the capability of the deep reinforcement learning algorithm and that there is room for improvment and investigation.

8.1 Limitations

One of the limitation of the present project is the reduced nature of the musculoskeletal model in terms of muscles and degrees of freedom. This restricts the algorithm to use only the existing muscles for both flexion and extension hence trading off the simplicity of optimization to the flexibility of complex model. Another limitation is the lack of in-built control architecture for processing the fluctuations and irregularities in actuator torques. These fluctuation considerably deteriorate the quality of the actuators and consequently, the gait. Since the coordination depends on both the legs, the deterioration of the quality in prosthetic leg could also hinder the quality of the movement of healthy leg.

Regarding the rewards, the imitation reward is referred from the imitation data of healthy individuals whereas the algorithm tries to build a policy for a prosthesis model based on this imitation data of healthy humans. This restricts the policy in attaining a flexible prosthesis locomotion and is forced to

follow the imitation data. Even though the imitation data has its advantages of guiding the policy towards an optimal policy, it also imposes a strict restrictions in flexibility to change the gait according to the actuators needs. Additionally, the weights involved in the reward shaping rewards are found to be notoriously hard to tune since the underlying effects of such tuning on neighboring muscles is not clear from quantitative or qualitative analysis.

Finally, it is computationally very expensive to train the deep reinforcement learning despite of paralell processing which puts a rigid time constraints on the amount of experiments that can be run. The iterations of the experiment can indeed be limited with techniques such as aearly stopping which comes with a tradeoff of losing any potential improvement of the policy in a long run.

8.2 Future outlook

Further investigation with full fledged model of more muscles can be conducted to get better accurate locomotion analysis. It is noted that a model with more muscles needs higher computational resources and complex algorithms to ensure the convergence of the problem and to avoid unrealistic motor behavior. Another potential improvement deals with feeding back the evaluation metrics such as symmetry angle and metabolic cost back to the reinforcement learning algorithm to enhance the coherence with the muscle states and environment. In terms of the actuators, the fluctuations can be smoothened inherently by developing a feedback loop to the reinforcement algorithm with additional incentive or penalty to ensure a constant steady torque. Another method to decrease the fluctuations is to use dynamic equations of the actuators to follow the steady increase and decrease in forces. The use of explicit dynamic equations of locomotion as heirarchical levels can further be integrated into the algorithm to converge the policy for an optimal gait. Furthermore, the features given to the deep neural network can be reconstructed with embeddings from dimensionality reduction which can result in faster convergence and utilize less computation power. Lastly, a custom loss function for the neural network optimization target at getting smoother curves can be adapted.

9 Conclusion

Deep Reinforcement learning coupled with bioinspired reward reshaping strategies to study locomotion of a prosthetic model in 3D environment for normal walking. The obtained outcomes suggest that computational resources, as well as domain knowledge of biomechanics and control theory are needed together to develop and evaluate an efficient and robust RL solution. The prosthetic model achieved convergence to a reasonable gait confirming the validity of the transfemoral prosthetic model. This also proves that the DRL can be used to design the control architectures of such models. As perspectives, current method can be extended by integrating the evaluation metrics used in the project into the algorithm to enhance the performance. The forces computed for actuators by the neural network shows an erratic pattern hence it cannot be directly used as control inputs in the control architecture of the transfemoral prosthesis. Furture work will focus on generating stable forces for the actuators that can be directly used as control of transfemoral prosthesis. Furthermore, in a grander scheme of things, deep reinforcement learning model can be investigated in the future in scope with the uncertainties of the musculoskeletal model and associated environment toward a general artificial intelligence solution for human locomotion learning.

Bibliography

[1] M. L. Handford and M. Srinivasan, "Robotic lower limb prosthesis design through simultaneous computer optimizations of human and prosthesis costs," *Scientific reports*, vol. 6, no. 1, pp. 1–7, 2016.

[2] W. Si, S.-H. Lee, E. Sifakis, and D. Terzopoulos, "Realistic biomechanical simulation and control of human swimming," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 1, pp. 1–15, 2014.

[3] F. De Groote and A. Falisse, "Perspective on musculoskeletal modelling and predictive simulations of human movement to assess the neuromechanics of gait," *Proceedings of the Royal Society B*, vol. 288, no. 1946, p. 20202432, 2021.

[4] 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, pp. 1–11, 2013.

[5] C. F. Ong, T. Geijtenbeek, J. L. Hicks, and S. L. Delp, "Predicting gait adaptations due to ankle plantarflexor muscle weakness and contracture using physics-based musculoskeletal simulations," *PLoS computational biology*, vol. 15, no. 10, p. e1006993, 2019.

[6] E. C. Ranz, J. M. Wilken, D. A. Gajewski, and R. R. Neptune, "The influence of limb alignment and transfemoral amputation technique on muscle capacity during gait," *Computer methods in BiomeChaniCs and BiomediCal engineering*, vol. 20, no. 11, pp. 1167–1174, 2017.

[7] V. J. Harandi, D. C. Ackland, R. Haddara, L. E. C. Lizama, M. Graf, M. P. Galea, and P. V. S. Lee, "Gait compensatory mechanisms in unilateral transfemoral amputees," *Medical engineering & physics*, vol. 77, pp. 95–106, 2020.

[8] Y. Lee, S. Kim, and J. Lee, "Data-driven biped control," in *ACM SIGGRAPH 2010 papers*, pp. 1–8, 2010.

[9] L. Liu, M. V. D. Panne, and K. Yin, "Guided learning of control graphs for physics-based characters," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 3, pp. 1–14, 2016.

[10] T. Kwon and J. K. Hodgins, "Momentum-mapped inverted pendulum models for controlling dynamic human motions," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 1, pp. 1–14, 2017.

[11] M. De Lasa, I. Mordatch, and A. Hertzmann, "Feature-based locomotion controllers," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4, pp. 1–10, 2010.

[12] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–14, 2018.

[13] W. Yu, G. Turk, and C. K. Liu, "Learning symmetric and low-energy locomotion," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–12, 2018.

[14] S. Coros, P. Beaudoin, and M. Van de Panne, "Generalized biped walking control," *ACM Transactions On Graphics (TOG)*, vol. 29, no. 4, pp. 1–9, 2010.

[15] I. Mordatch, M. De Lasa, and A. Hertzmann, "Robust physics-based locomotion using low-dimensional planning," in *ACM SIGGRAPH 2010 papers*, pp. 1–8, 2010.

[16] N. Heess, D. TB, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S. Eslami, *et al.*, "Emergence of locomotion behaviours in rich environments," *arXiv preprint arXiv:1707.02286*, 2017.

[17] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[18] Ł. Kidziński, C. Ong, S. P. Mohanty, J. Hicks, S. Carroll, B. Zhou, H. Zeng, F. Wang, R. Lian, H. Tian, *et al.*, "Artificial intelligence for prosthetics: Challenge solutions," in *The NeurIPS'18 Competition*, pp. 69–128, Springer, 2020.

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

[20] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–35, 2017.

[21] S. L. Delp, F. C. Anderson, A. S. Arnold, P. Loan, A. Habib, C. T. John, E. Guendelman, and D. G. Thelen, "Opensim: open-source software to create and analyze dynamic simulations of movement," *IEEE transactions on biomedical engineering*, vol. 54, no. 11, pp. 1940–1950, 2007.

[22] 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.

[23] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Icml*, vol. 99, pp. 278–287, 1999.

[24] A. D. Laud, *Theory and application of reward shaping in reinforcement learning*. University of Illinois at Urbana-Champaign, 2004.

[25] S. Lee, M. Park, K. Lee, and J. Lee, "Scalable muscle-actuated human simulation and control," *ACM Transactions On Graphics (TOG)*, vol. 38, no. 4, pp. 1–13, 2019.

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

[27] K. M. Steele, A. Seth, J. L. Hicks, M. S. Schwartz, and S. L. Delp, "Muscle contributions to support and progression during single-limb stance in crouch gait," *Journal of biomechanics*, vol. 43, no. 11, pp. 2099–2105, 2010.

[28] F. Dzeladini, J. Van Den Kieboom, and A. Ijspeert, "The contribution of a central pattern generator in a reflex-based neuromuscular model," *Frontiers in human neuroscience*, vol. 8, p. 371, 2014.

[29] S. Aoi, T. Ohashi, R. Bamba, S. Fujiki, D. Tamura, T. Funato, K. Senda, Y. Ivanenko, and K. Tsuchiya, "Neuromusculoskeletal model that walks and runs across a speed range with a few motor control parameter changes based on the muscle synergy hypothesis," *Scientific reports*, vol. 9, no. 1, pp. 1–13, 2019.

[30] F. De Groote, A. Van Campen, I. Jonkers, and J. De Schutter, "Sensitivity of dynamic simulations of gait and dynamometer experiments to hill muscle model parameters of knee flexors and extensors," *Journal of biomechanics*, vol. 43, no. 10, pp. 1876–1883, 2010.

[31] F. De Groote, A. L. Kinney, A. V. Rao, and B. J. Fregly, "Evaluation of direct collocation optimal control problem formulations for solving the muscle redundancy problem," *Annals of biomedical engineering*, vol. 44, no. 10, pp. 2922–2936, 2016.

[32] T. K. Uchida, A. Seth, S. Pouya, C. L. Dembia, J. L. Hicks, and S. L. Delp, "Simulating ideal assistive devices to reduce the metabolic cost of running," *PloS one*, vol. 11, no. 9, p. e0163417, 2016.

[33] A. Falisse, G. Serrancolí, C. L. Dembia, J. Gillis, I. Jonkers, and F. De Groote, "Rapid predictive simulations with complex musculoskeletal models suggest that diverse healthy and pathological human gaits can emerge from similar control strategies," *Journal of The Royal Society Interface*, vol. 16, no. 157, p. 20190402, 2019.

[34] G. Li, R. Gomez, K. Nakamura, and B. He, "Human-centered reinforcement learning: A survey," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 4, pp. 337–349, 2019.

[35] J. Hua, L. Zeng, G. Li, and Z. Ju, "Learning for a robot: Deep reinforcement learning, imitation learning, transfer learning," *Sensors*, vol. 21, no. 4, p. 1278, 2021.

[36] M. H. Schwartz, A. Rozumalski, and J. P. Trost, "The effect of walking speed on the gait of typically developing children," *Journal of biomechanics*, vol. 41, no. 8, pp. 1639–1650, 2008.

[37] A. C. Tenorio-Gonzalez, E. F. Morales, and L. Villasenor-Pineda, "Dynamic reward shaping: training a robot by voice," in *Ibero-American conference on artificial intelligence*, pp. 483–492, Springer, 2010.

[38] P. Mannion, S. Devlin, J. Duggan, and E. Howley, "Reward shaping for knowledge-based multi-objective multi-agent reinforcement learning," *The Knowledge Engineering Review*, vol. 33, 2018.

[39] Y. Hu, W. Wang, H. Jia, Y. Wang, Y. Chen, J. Hao, F. Wu, and C. Fan, "Learning to utilize shaping rewards: A new approach of reward shaping," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15931–15941, 2020.

[40] S. Devlin and D. Kudenko, "Theoretical considerations of potential-based reward shaping for multi-agent systems," in *The 10th International Conference on Autonomous Agents and Multiagent Systems*, pp. 225–232, ACM, 2011.

[41] A. Hussein, E. Elyan, M. M. Gaber, and C. Jayne, "Deep reward shaping from demonstrations," in *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 510–517, IEEE, 2017.

[42] S. Doncieux, "Transfer learning for direct policy search: A reward shaping approach," in *2013 IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, pp. 1–6, IEEE, 2013.

[43] G. Vasan and P. M. Pilarski, "Learning from demonstration: Teaching a myoelectric prosthesis with an intact limb via reinforcement learning," in *2017 International Conference on Rehabilitation Robotics (ICORR)*, pp. 1457–1464, IEEE, 2017.

[44] 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, pp. 1–13, 2017.

[45] Z. Xie, G. Berseth, P. Clary, J. Hurst, and M. van de Panne, "Feedback control for cassie with deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1241–1246, IEEE, 2018.

[46] 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)*, pp. 537–543, IEEE, 2019.

[47] J. Weng, E. Hashemi, and A. Arami, "Natural walking with musculoskeletal models using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 4156–4162, 2021.

[48] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[49] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[50] M. Riedmiller, "Neural fitted q iteration–first experiences with a data efficient neural reinforcement learning method," in *European conference on machine learning*, pp. 317–328, Springer, 2005.

[51] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh, "Action-conditional video prediction using deep networks in atari games," *Advances in neural information processing systems*, vol. 28, 2015.

[52] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in Neural Information Processing Systems* (S. Solla, T. Leen, and K. Müller, eds.), vol. 12, MIT Press, 1999.

[53] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[54] C. T. John, F. C. Anderson, J. S. Higginson, and S. L. Delp, "Stabilisation of walking by intrinsic muscle properties revealed in a three-dimensional muscle-driven simulation," *Computer methods in biomechanics and biomedical engineering*, vol. 16, no. 4, pp. 451–462, 2013.

[55]

[56] A. Esquenazi, "Gait analysis in lower-limb amputation and prosthetic rehabilitation," *Physical Medicine and Rehabilitation Clinics*, vol. 25, no. 1, pp. 153–167, 2014.

[57] R. Andres and S. Stimmel, "Prosthetic alignment effects on gait symmetry: a case study," *Clinical biomechanics*, vol. 5, no. 2, pp. 88–96, 1990.

[58] S. Viteckova, P. Kutilek, Z. Svoboda, R. Krupicka, J. Kauler, and Z. Szabo, "Gait symmetry measures: A review of current and prospective methods," *Biomedical Signal Processing and Control*, vol. 42, pp. 89–100, 2018.

[59] M. Błażkiewicz, I. Wiszomirska, and A. Wit, "Comparison of four methods of calculating the symmetry of spatial-temporal parameters of gait," *Acta of bioengineering and biomechanics*, vol. 16, no. 1, 2014.

[60] R. A. Zifchock, I. Davis, J. Higginson, and T. Royer, "The symmetry angle: a novel, robust method of quantifying asymmetry," *Gait & posture*, vol. 27, no. 4, pp. 622–627, 2008.

[61] F. E. Zajac, R. R. Neptune, and S. A. Kautz, "Biomechanics and muscle coordination of human walking: part ii: lessons from dynamical simulations and clinical implications," *Gait & posture*, vol. 17, no. 1, pp. 1–17, 2003.

[62] L. J. Bhargava, M. G. Pandy, and F. C. Anderson, "A phenomenological model for estimating metabolic energy consumption in muscle contraction," *Journal of biomechanics*, vol. 37, no. 1, pp. 81–88, 2004.

[63] A. Minetti and R. M. Alexander, "A theory of metabolic costs for bipedal gaits," *Journal of Theoretical Biology*, vol. 186, no. 4, pp. 467–476, 1997.

[64] H. Houdijk, M. Bobbert, and A. De Haan, "Evaluation of a hill based muscle model for the energy cost and efficiency of muscular contraction," *Journal of biomechanics*, vol. 39, no. 3, pp. 536–543, 2006.

[65] A. D. Koelewijn, D. Heinrich, and A. J. van den Bogert, "Metabolic cost calculations of gait using musculoskeletal energy models, a comparison study," *PloS one*, vol. 14, no. 9, p. e0222037, 2019.

[66] J. H. Kim and D. Roberts, "A joint-space numerical model of metabolic energy expenditure for human multibody dynamic system," *International Journal for Numerical Methods in Biomedical Engineering*, vol. 31, no. 9, p. e02721, 2015.

[67] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.

[68] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

[69] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283, 2016.

[70] L. Dalcin and Y.-L. L. Fang, "Mpi4py: Status update after 12 years of development," *Computing in Science & Engineering*, vol. 23, no. 4, pp. 47–54, 2021.

[71] OpenAI, "Openai five." `https://blog.openai.com/openai-five/`, 2018.

[72] J. Zhu, Y. Wang, J. Jiang, B. Sun, and H. Cao, "Unidirectional variable stiffness hydraulic actuator for load-carrying knee exoskeleton," *International Journal of Advanced Robotic Systems*, vol. 14, p. 172988141668695, 02 2017.

[73] J. Realmuto, G. Klute, and S. Devasia, "Nonlinear passive cam-based springs for powered ankle prostheses," *Journal of Medical Devices*, vol. 9, 03 2015.