NeuroProsthetics

Thijs Lukkien (s3978389) Jiri Derks (s4003039)

March 2024

1 Inverse Kinematics and Dynamics

Question 1

a) Explain what an inverse kinematic simulation is and how it is performed.

Inverse kinematics refers to modeling a physical body based on real-life measurements. This allows you to create and align a model from a set of markers placed on a human body. You can use two types of marker sets:

- anatomical reference frames; this method uses markers placed on anatomically recognizable places (for example on top of the knee), also called 'anatomical landmarks'.
- Tracking reference frames; this method uses markers placed on body segments that are not necessarily linked to anatomical reference points. These markers require three noncolinear markers on each segment, which are placed on soft tissue with small motion. These markers are placed such that they are visible for a majority of the time.

For either method, a physical (often skeletal) body model needs to be created in the simulation. Markers are placed on a real-world human body and replicated on the simulated model. The physical and simulated markers are fitted together by minimizing the sum of all distances between each pair of markers. The measurements of the real-life markers are then mapped onto the markers on the simulation model.

b) Explain how inverse kinematic simulation is done in OpenSim and explain the decisions you made on the parameters you set in the simulation.

As already explained in the previous question, real-life and simulated markers are mapped onto each other. This way, movement data can be converted into moving points in a simulation. When this is connected to a skeletal system, it can simulate a walking body.

For the inverse kinetics simulation, we decided to skip the first gait (t=19.0 until t=20.285) in order to use the most 'fluid' movements. Hence, we ended up using the second, third, and fourth gaits the times of which can be found in Table 1.

Gait number	**Start Time**	**Duration**
1	20.285	1.095
2	21.380	1.055
3	22.435	1.09

Table 1: The starting time and duration of the chosen gaits for the inverse kinemmatics

c) Report the root mean square errors (RMSEs) between the virtual makers of the scaled model and the experimental markers in the Camargo's dataset.

When applying the scaling methods described in Tutorial 1, the root mean squared error is ≈ 0.0384m (or $\approx$3.84cm).

d) Change the "static pose weight" of the virtual markers and reflect on why the RMSEs reduce or increases.

In our opinion, anatomical markers are more important than tracking reference markers (markers placed on flesh) as they are more accurate (by being directly on bone) and thus form a closer relation between the skeletal simulation model and the real-world markers. For this reason, we increased the static pose weights of anatomical markers to 1.5 and decreased the static pose weights of tracking reference markers to 0.5. The resulting RMSE is 0.0398471. This score is slightly worse than the original setting, though it may also result in the important simulation markers fitting better to real-world data.

e) Plot the ankle/knee/hip angles of 3 gaits (mean and standard deviation) over a gait cycle (0-100%). Explain the plots also in reference with the plots of healthy subjects that can be found in the literature. Report the RMSE of the inverse kinematics in the last time step and explain its meaning.

Figures 1-6 depict the angles of the knees, the ankles, and the hip flexion over 3 gaits.

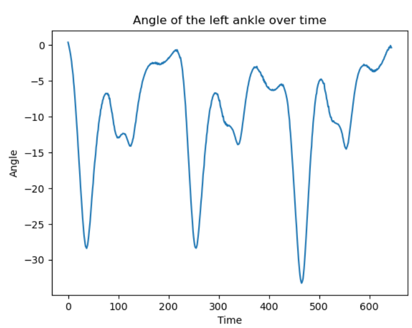

Figure 1: Plot depicting the angle of the left ankle.

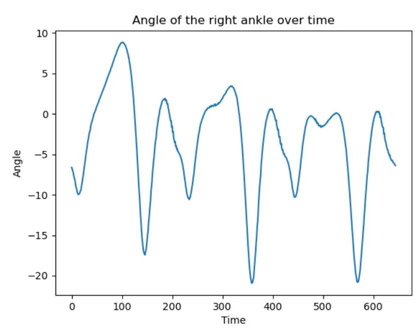

Figure 2: Plot depicting the angle of the right ankle.

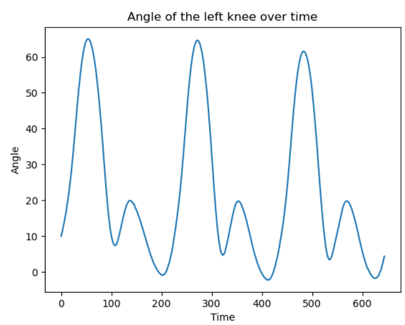

Figure 3: Plot depicting the angle of the left knee.

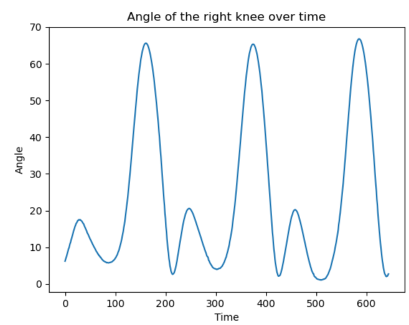

Figure 4: Plot depicting the angle of the right knee.

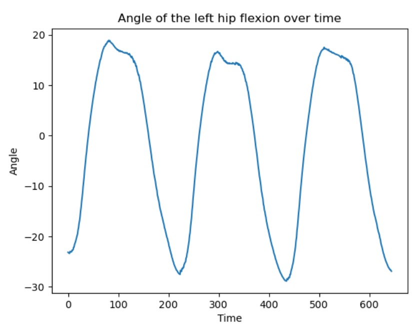

Figure 5: Plot depicting the angle of the left hip flexion.

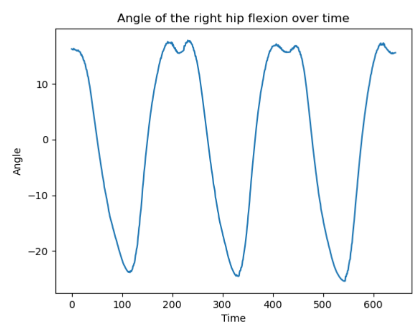

Figure 6: Plot depicting the angle of the right hip flexion.

When comparing this to the plots described in the literature we see that the angles of the knees and hips are very similar to those described in the literature.

However, there is a noticeable deviation in the angles of the left ankle from the literature. While the general shape of the left ankle plot mirrors the literature, the angles are shifted into the negative range.

Question 2

a) Explain what an inverse dynamics simulation is and how it is performed

Inverse dynamics focuses on determining the forces and torques in a given system, in this case, the musculoskeletal model. This is done by using marker data to construct the kinematics of the model, after obtaining the kinematics of the model an inverse dynamics solver can be used to determine the forces and torques needed to perform the motions of the kinematic model.

In this exercise, ground reaction forces are used to calculate the forces of the muscles.

Thus inverse kinematics can be seen as the study of the angles between the bones, and inverse dynamics can be seen as the study of the muscle forces.

b) Explain how inverse dynamics simulation is done in OpenSim and explain the decisions you made on the parameters you set in the simulation.

In OpenSim, we used the scaled model gained earlier. We could then use the inverse dynamics tool. This calculates the torques and forces required to make the observed movements as discussed in the previous question.

Here, we used the motion data obtained from the inverse kinematics as input. The coordinates were filtered with a frequency of 6 Hz. Lastly, we loaded the external forces applied to our system as well.

c) Plot the ankle/knee/hip torques of 1 gait over a gait cycle (0-100Explain the plots also in reference with the plots of healthy subjects that can be found in the literature. Note that certain limitations of your simulation depend on the used dataset.

In Figures 7-12 we have plotted the torques of the ankles, knees, and hips. We see that the angles found in the inverse dynamics are very similar to those found in the literature. The knee angles both have a small peak followed by a larger peak for the gait cycle, which aligns with the literature. Similarly, the hip flexion has one large valley followed by a fairly straight peak (that has a slight dip at the peak), also following the data in the literature. The ankles are a bit more erratic in this data, the overall shape does match that of the data in the literature, however, the ankles' first valley is larger than that seen in the training data.

Figure 7: Plot depicting the angle of the left ankle.

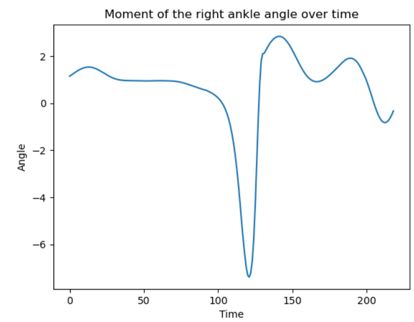

Figure 8: Plot depicting the angle of the right ankle.

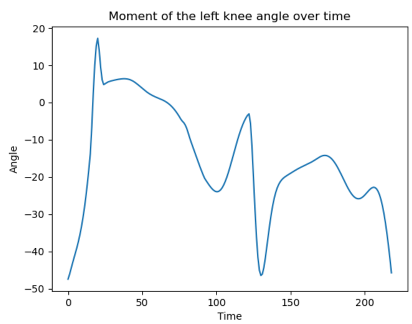

Figure 9: Plot depicting the angle of the left knee.

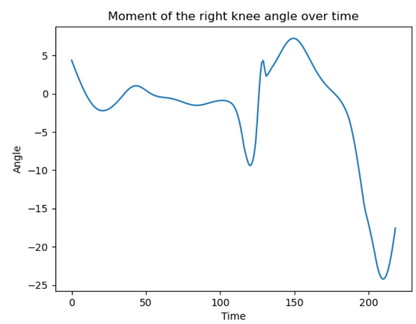

Figure 10: Plot depicting the angle of the right knee.

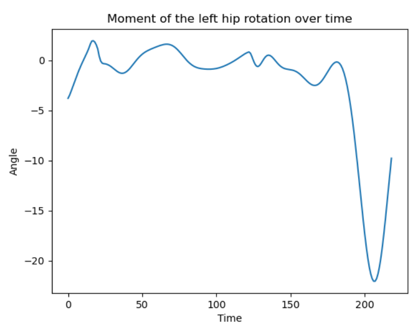

Figure 11: Plot depicting the angle of the left knee.

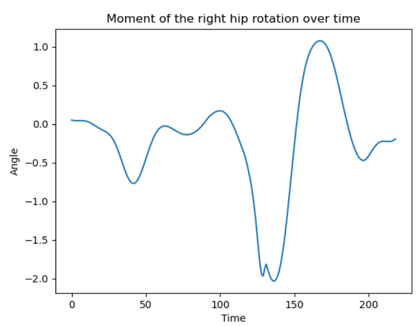

Figure 12: Plot depicting the angle of the right knee.

d) Plot the forces of the 22 muscles of 1 gait over a gait cycle (0-100you can plot the raw data and fit a polynomial function. Explain how they were obtained and discuss whether they relate to the plots of healthy subjects that can be found in the literature. Note that certain limitations of your simulation depend on the used dataset.

The following plots were obtained by running the inverse dynamics in opensim. This produced a forces file, in which the forces of the muscles were plotted over time. This data was extracted and plotted. For each plot, a polynomial was fitted as a way to smooth the data.

Comparing this data we again see that the overall shapes match the literature pretty well. However, the shape of the ankles moment seems to not match up with the literature very well, as the slope of the peaks seem to be higher than those in the literature. For the other movements (the knees and the hips) the data seems in line with the literature.

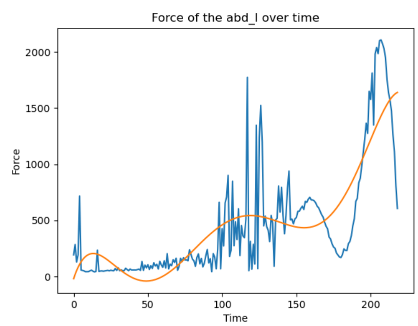

Figure 13: Plot depicting the angle of the left ankle.

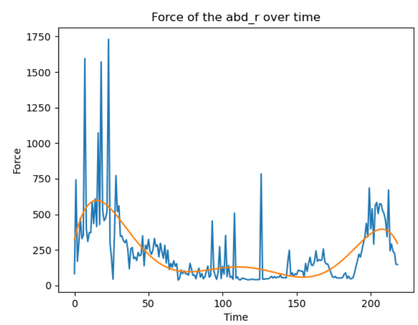

Figure 14: Plot depicting the angle of the right ankle.

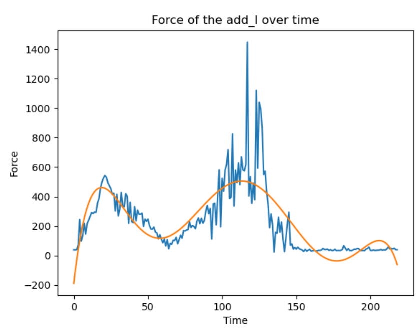

Figure 15: Plot depicting the angle of the left ankle.

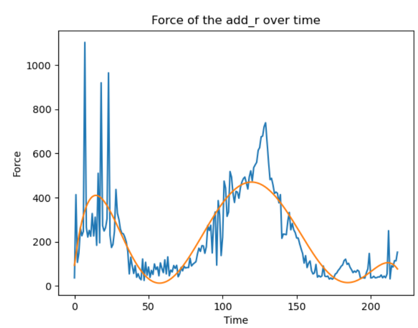

Figure 16: Plot depicting the angle of the right ankle.

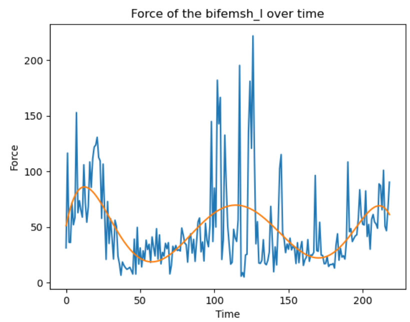

Figure 17: Plot depicting the angle of the left ankle.

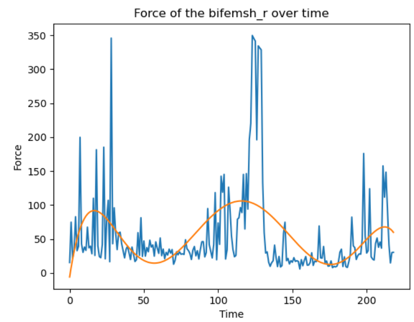

Figure 18: Plot depicting the angle of the right ankle.

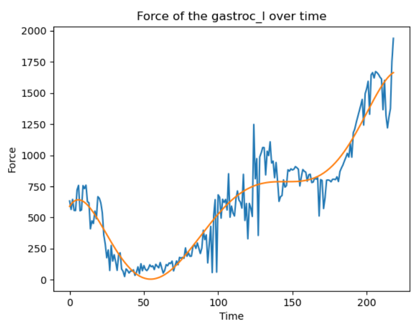

Figure 19: Plot depicting the angle of the left ankle.

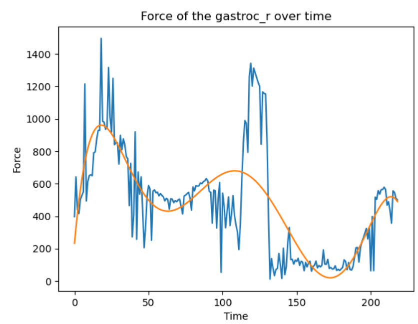

Figure 20: Plot depicting the angle of the right ankle.

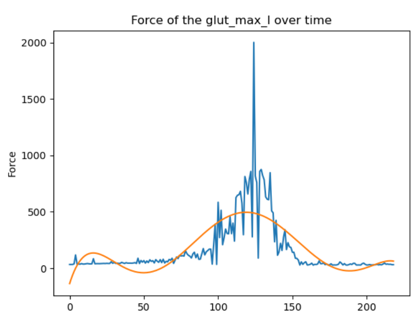

Figure 21: Plot depicting the angle of the left ankle.

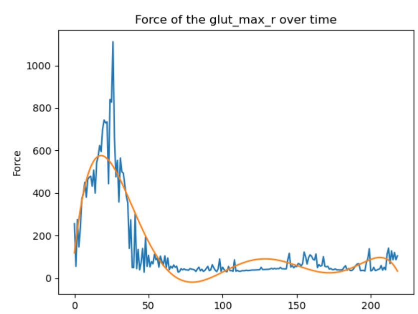

Figure 22: Plot depicting the angle of the right ankle.

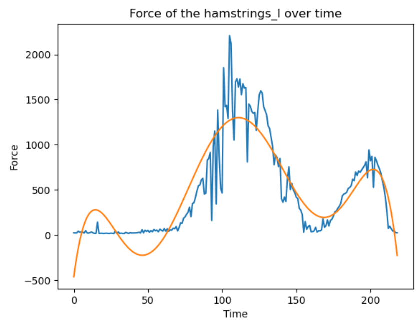

Figure 23: Plot depicting the angle of the left ankle.

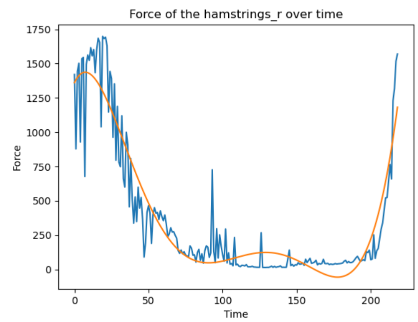

Figure 24: Plot depicting the angle of the right ankle.

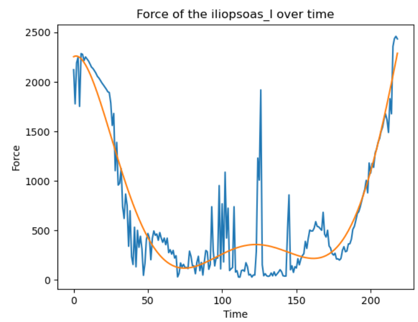

Figure 25: Plot depicting the angle of the left ankle.

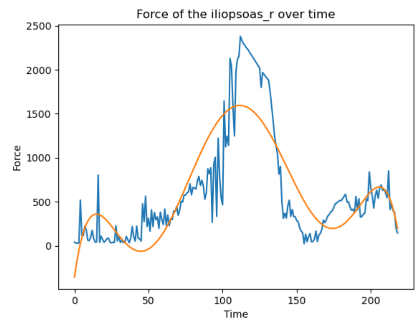

Figure 26: Plot depicting the angle of the right ankle.

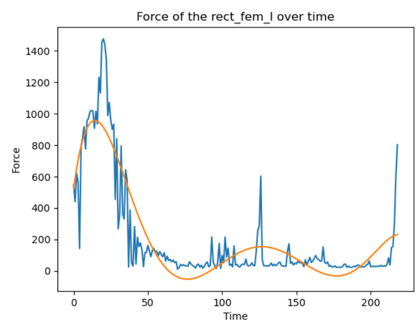

Figure 27: Plot depicting the angle of the left ankle.

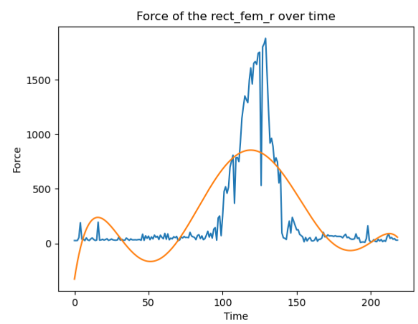

Figure 28: Plot depicting the angle of the right ankle.

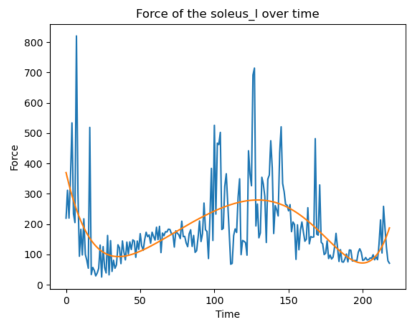

Figure 29: Plot depicting the angle of the left ankle.

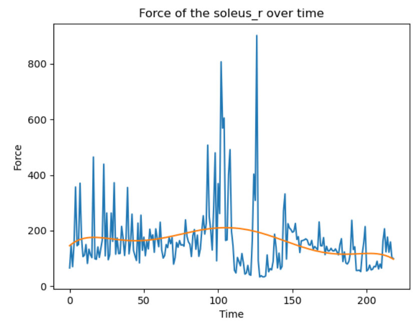

Figure 30: Plot depicting the angle of the right ankle.

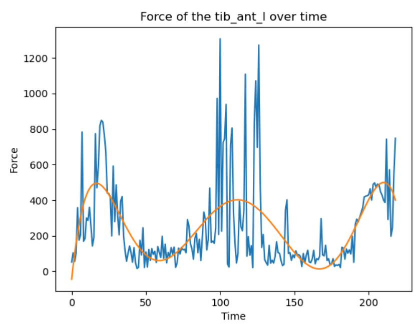

Figure 31: Plot depicting the angle of the left ankle.

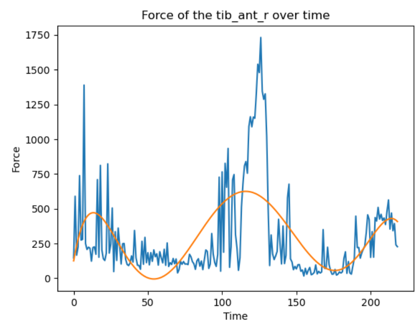

Figure 32: Plot depicting the angle of the right ankle.

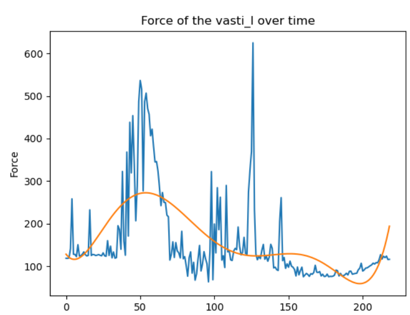

Figure 33: Plot depicting the angle of the left ankle.

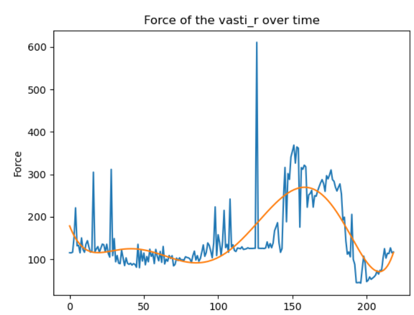

Figure 34: Plot depicting the angle of the right ankle.

2 Forward Dynamics

2.1 Introduction

For part 2 we were tasked with running forward dynamics using a reinforcement learning architecture. Along with this, we were also tasked with making a change to the architecture and discussing the improvements this change made to the model.

2.2 Methods

The forward dynamics simulaution was performed with Opensim using the gait1415+2 model, and reinforcement learning. We used proximal policy optimization (PPO), which was developed by openAI in 2017 (Schulman, Wolski, Dhariwal, Radford, & Klimov, 2017). Proximal Policy Optimization is a optimization technique that, unlike traditional deep-reinforcment learning using Q-leanring (DQN), tries to maximize a certain objective function using gradient ascent. PPO is based on trust region policy optimization (TRPO), which also tries to maximize an objective function and is more stable than DQN. TRPO utilizes a kl-divergence term, which indicates the difference between two sets of distributions (in this case the parameters of the new and old network in training), to make sure that during training the network is not updated to much and thus increase stability. However, TRPO is very complicated and hard to implement. Thus PPO is now often used due to it only using first-order optimization, and replaces the KL-diverence with clipping. PPO maximizes the following objective function:

$$L^{CLIP}(\theta) = \hat{E}_t[min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+epsilon)A_t)] \quad (1)$$

Where $\hat{A}_t$ is an estimator of the advantage function at timestep t, $\hat{E}_t$ mpirical average over a finite batch of samples, in an algorithm that alternates between sampling and optimization, and $r_t(\theta)$ is defined as follows:

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{oldl}}(a_t|s_t)} \quad (2)$$

Due to the clipping the model is incentivised to not move r_t outside the interval $[1-\epsilon, 1+\epsilon]$ and thus reduces the size of the steps the model can make while training.

The reward function consists of two components, the imitation reward, and the goal reward. The imitation reward focuses on the angles of the joints, and comparing them to the imitation data. Whereas the goal reward looks at the position of the pelvis of the model. The reward was calculated as follows:

$$reward = 0.9 * reward_{imitation} + 0.1 * reward_{goal} \quad (3)$$

The neural network consisted of 2 hidden layers of size = 128, the input layer was the size of the observation state which was 28 inputs:

- 6 positions and 6 velocities of the degrees of freedom (DOFs) of the pelvis
- 4 positions and 4 velocities of the DOFs of the two hips (2 DOFs per hip)
- 2 positions and 2 velocities of the DOFs of the two knees (1 DOF per knee)
- 2 positions and 2 velocities of the DOFs of the two ankles (1 DOF per ankle)

The output layer gave the activations for the 15 muscles and 2 actuators of the prosthetic.

the other parameters used for the training of the default model were:

Timesteps	30 000 000
number of environments	32
steps per iteration	6144
batch size	22.435
initial ϵ value	0.001
number of epochs	4
clip range	0.25

The change we implemented to the model described above was that of dynamically adjusting the reward for the agent. Initially, we trained the agent over 30 million timesteps, resulting in a decent performance. However, the agent was not able to consistently walk in a straight line and lost stability over time. Thus we believe that a form of transfer learning is suitable in this case. In transfer learning it is assumed that the knowledge gained of the model for one task can be used as a starting point for the training of a new model in a similar task. We could use the trained model as a starting point and decrease the reward for imitation and increase the reward for the goal (x, y, and z-positions) of the model. We hypothesized that the pre-trained model's weights would give a good base to be able to change the reward function to focus more on achieving a better goal reward and thus increasing the stability, while still walking in a human-like manner and retaining a normal gait cycle.

2.3 results

Overall 3 models were trained, one using the settings described in the methods, for 30M timesteps. Two models were trained using transfer learning, using the first model as a base. These models trained for an additional 30M timesteps each but converged to a policy after approximately 2M timesteps. This can be seen in Figure 35. These models used a starting learning rate of 0.00005 to ensure that the model would not deviate too much from the starting model, and forget the knowledge obtained through previous learning.

Figure 35: The episode length over time, *model 100*, and *model 50_50* were trained using *model default* as a base. The y-axis shows how long a model was able to keep its balance while walking in a straight line.

The second model (50_50), was trained using the new reward function in which the balance of the rewards was $reward = 0.5 * reward_{imitation} + 0.5 * reward_{goal}$. The third model (100) was trained with the reward function $reward = reward_{goal}$. The joint angles and velocities for these models can be seen in Figures 36, 37, and 38.

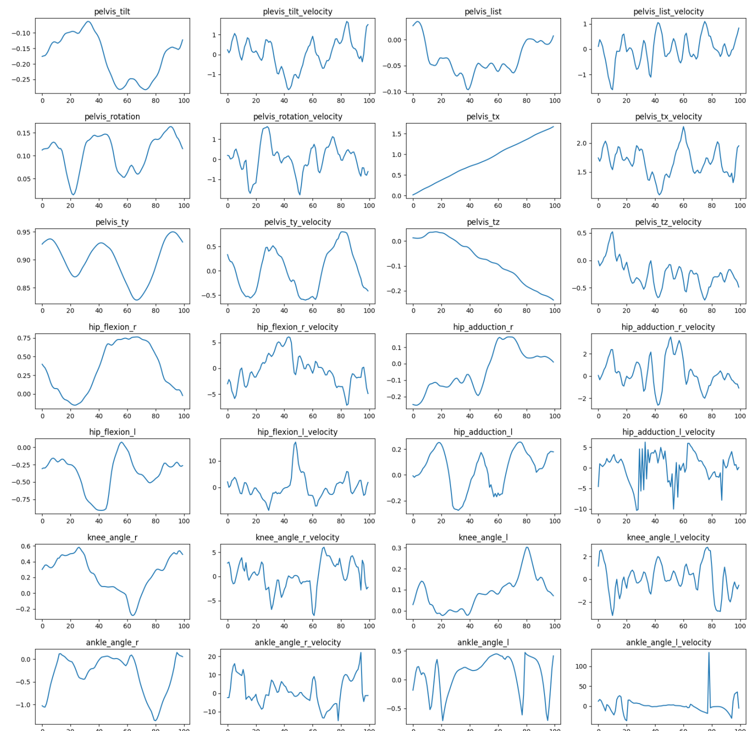

Figure 36: Plot depicting angles and velocity of the joints, and the positions of the pelvis of the default model

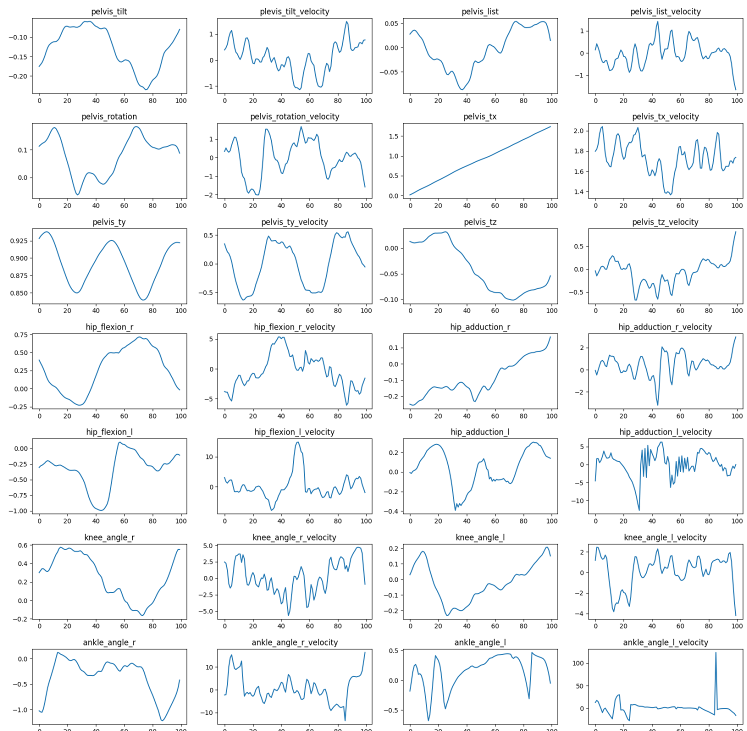

Figure 37: Plot depicting angles and velocity of the joints, and the positions of the pelvis of the 50_50 model.

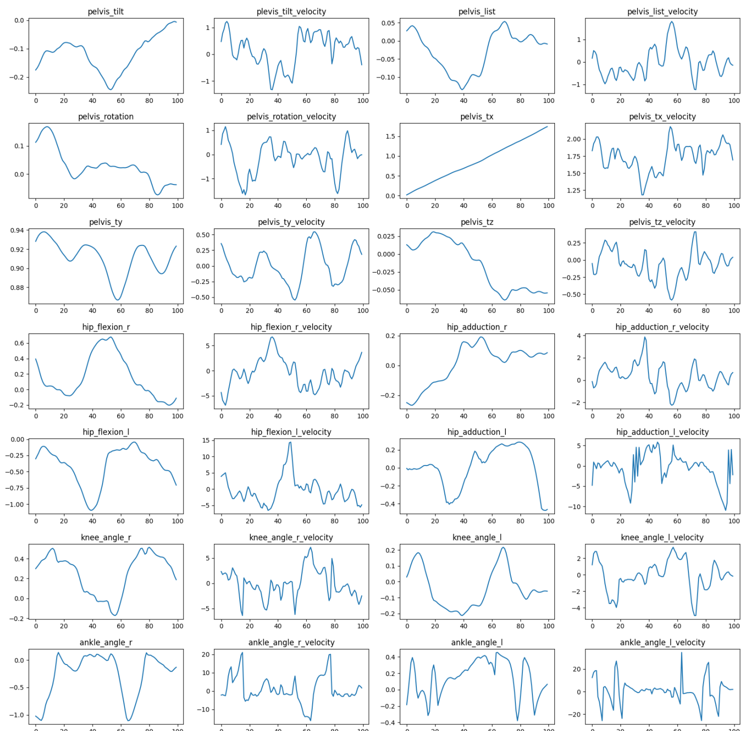

Figure 38: Plot depicting angles and velocity of the joints, and the positions of the pelvis of the 100 model

As we wanted to increase the stability of the model, we can look at the average z-deviation of the models. We see that the default model is likely to deviate, while the models trained further showed a reduced deviation. This is seen in Figures 39, 40, and 41. These plots show the average of the z-position of the pelvis over time, along with the standard deviation of these runs. 100 runs were performed to gather this data. (Note that some runs did not make the full amount of timesteps, leading to the data at the end of the plots acting somewhat erratically.)

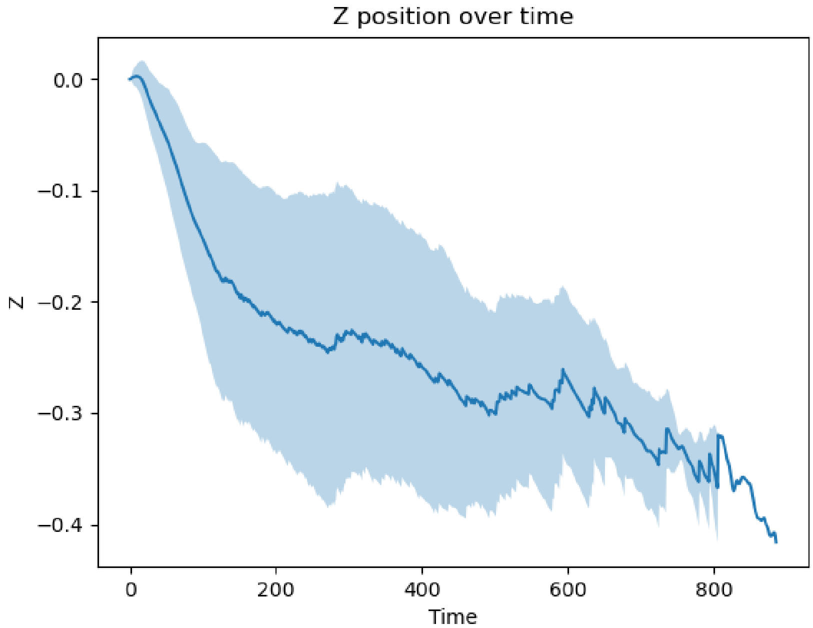

Figure 39: Plot depicting z-positions of the pelvis of the default model, averaged over 100 runs. The blue area shows the standard deviation over these 100 runs. One can read from this plot that the model always falls to one side over 800 timesteps.

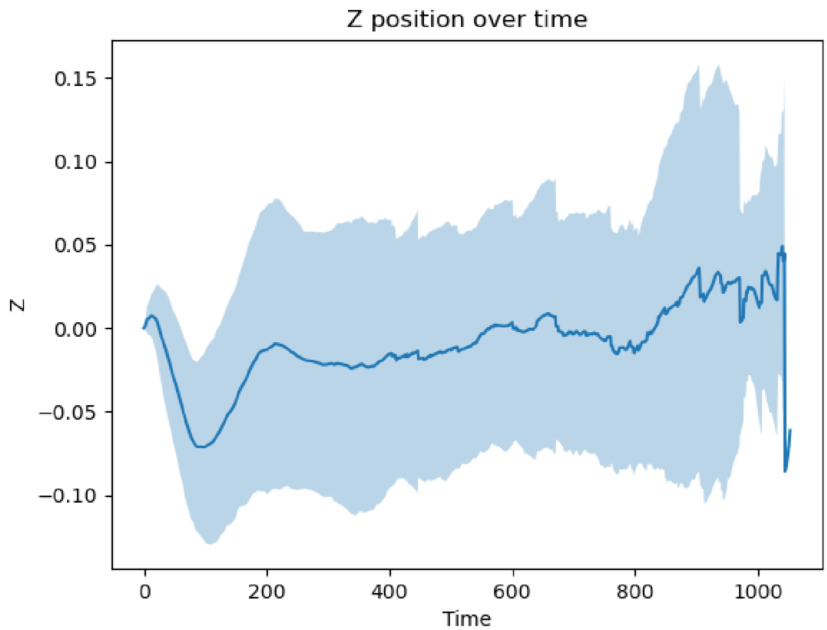

Figure 40: Plot depicting z-positions of the pelvis of the 50_50 model, averaged over 100 runs. The blue area shows the standard deviation over these 100 runs.

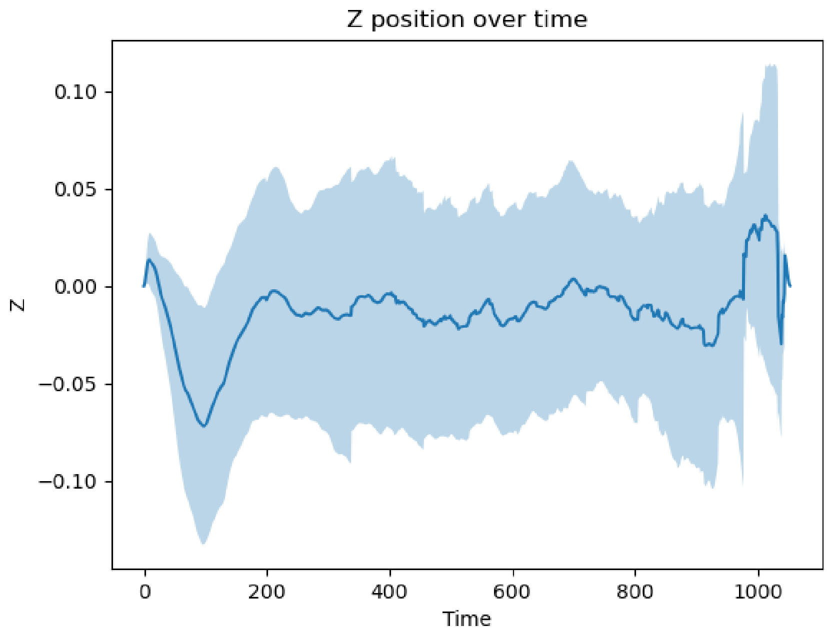

Figure 41: Plot depicting z-positions of the pelvis of the 100 model, averaged over 100 runs. The blue area shows the standard deviation over these 100 runs.

The muscle forces can be seen in Figures 42, 43, and 44. These figures show the muscle activations of the reinforcement learning model over a gait cycle. The activations are discrete, so a red line is plotted over the activations showing a running average with a window of 10 timesteps.

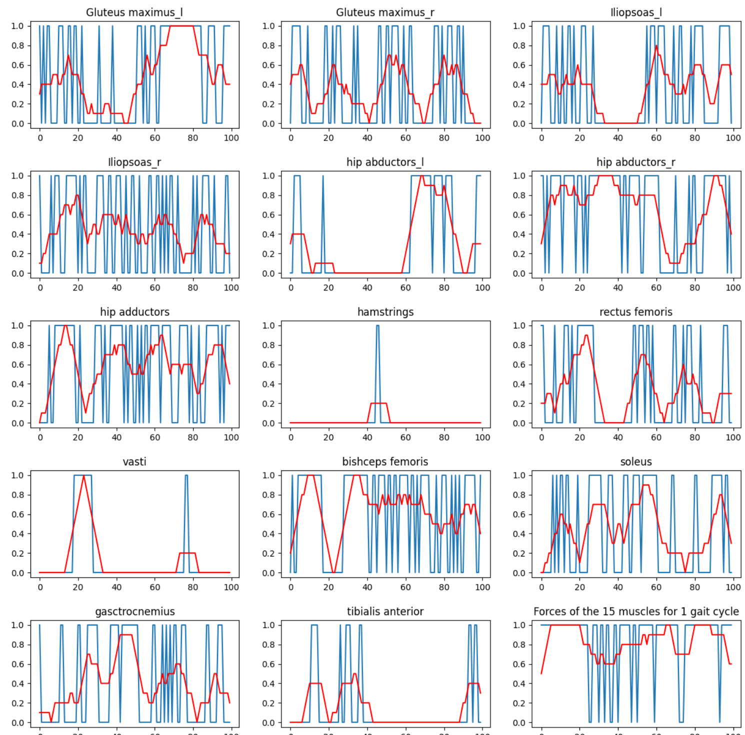

Figure 42: Plots depicting the muscle and actuator forces of the default model

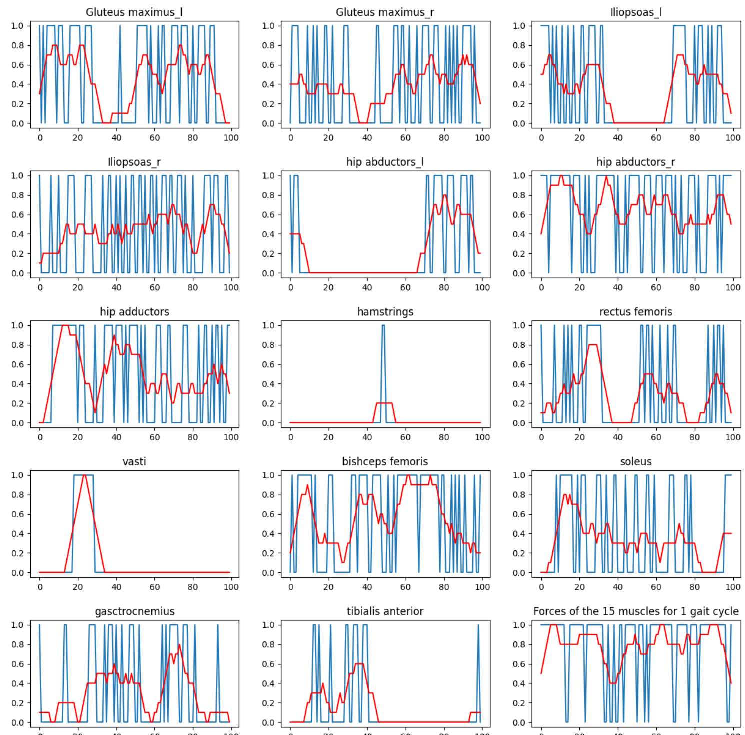

Figure 43: Plots depicting the muscle and actuator forces of the 50_50 model

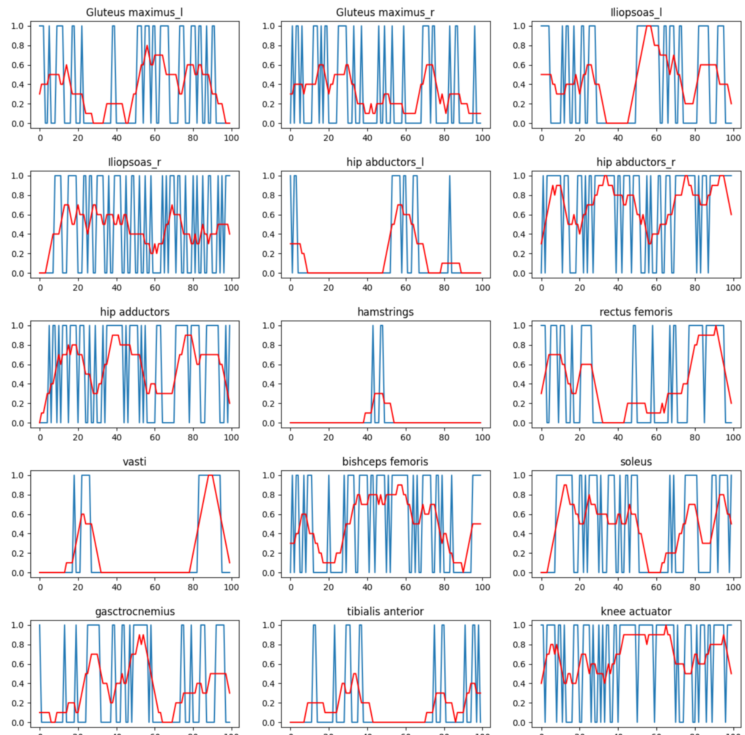

Figure 44: Plot depicting the muscle and actuator forces of the 100 model

2.4 Discussion

In Figures 39, 40, and 41, we see that the stability in the z-axis increases for the models with the adjusted reward function. Furthermore, in Figure 35 we can see that the models trained with these adjusted reward functions converge to a new optimal policy fairly quickly, only using approximately 2M timesteps to converge. We also see that the new policy learned is more stable than the old policy. However, this may be due to the fact that the models start learning using a lower learning rate. In the plots depicting the model's deviation from the x-axis (deviation in z-coordinate), we see that there is little difference in means between the model that uses 100% goal reward and the model in which the reward is split evenly. However, we can see that the standard deviation of this z-deviation is twice as large in the 50_50 model compared to the deviation in the 100 model. This seems to suggest that the model with 100 % goal reward is able to walk straight in more runs than the 50_50 model.

When looking at the simulation visually this model does indeed seem to deviate from the x-axis less. However, it does this while performing some weird walking behavior, where the model can cross their legs while walking. As there is no self-collision in the model the agent can move one of its legs through another leg. There are some instances of this behavior in the default model, however this occurs very infrequently. The 50_50 model also does not seem to cross its legs in this manner frequently and is thus in our opinion the better option.

When comparing the gait cycles of the different models, we see that overall most graphs seem to stay similar over the 3 different models, this is a good indication that the model is still able to walk normally with the different reward functions. Although they seem similar, there are a few notable differences in the gait cycles of the different models. The first is that the pelvis rotation becomes less pronounced the more goal reward is added to the overall goal. We also see that in all the reinforcement learning models, the left ankle (which is the prosthetic) performs quite erratic when compared to the ankle's angles found in the literature. This can be explained by the fact that the left ankle's activations are different to the other muscles, as these are actuators and not muscles. It would be interesting for further research to make this actuator behave in a manner more similar to those of normal muscles, to achieve better gait cycles.

3 Bonus question

In question 1d (part 2), we showed how transfer learning can be applied to improve model performance. There, we implemented transfer learning using a new reward function. We hypothesize that this direction holds a lot more potential for further improvements. One direction that we are curious to explore is to use transfer learning between participants. The system of walking can be distilled down into a model, and by combining multiple sources of data, hence transfer training a model between participants, we hope to encode more knowledge about moving into one model than we currently can with one participant.

We have been unable to implement this, as we are limited by time constraints, although we believe this may be the best way of further improving the walking model.

One possible strategy to implement this is to change the reward function gradually over time, rather than at one point in time, as we have done now. For example, in iteration 0 of training the model would use 90% imitation reward and 10% goal reward, and in iteration 5M it could be 80% imitation and 20% goal reward. This would allow the model to slowly adjust to its new goal so that it becomes less likely to spontaneously develop strange walking patterns as was observed for the 100 model.

4 Personal contributions

We split the work equally and both of us worked on each part. Jiri focussed some more on the simulation while Thijs focussed mainly on the theoretical questions.

References

Bozinovski, S. (2020). Reminder of the first paper on transfer learning in neural networks, 1976. *Informatica*, *44*(3).

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.