Evaluating Optimization Algorithms for Human Gait Simulations

Human gait can be simulated by solving optimal control problems. This process replicates the optimization process that the central nervous system solves when generating movements. Such simulations of human gait have different purposes: they can be used to predict the effect of assistive devices like exoskeletons [1,2] or prosthesis [3], or running shoes [4] on gait performance, while they can also be used to reconstruct recorded movements from different types of data, such as data from the gold standard of optical motion capture and force plates [5], inertial sensor measurements [6-8], and radar [9].

The optimal control problem is commonly solved using direct collocation with a backward Euler discretization, converting the problem into a large-scale nonlinear optimization problem. This problem is then solved with a gradient-based optimization algorithm, usually IPOPT [10]. However, several other optimization algorithms exist, while there are also internal functions of IPOPT, such as the linear solver, that could be set.

The aim of this thesis is to investigate the performance of different optimization algorithms for these human gait simulations. These simulations can be very time consuming, especially when three dimensional human models are used. Furthermore, due to the inherent non-convexity of the problem, there is a risk of solving to local minima that are not realistic, especially when using inertial sensor data. Therefore, we aim to investigate the settings of the optimization algorithm to ensure that proper simulations are found as time efficiently as possible. The this will make use of an open-source toolbox that has been developed to create human gait simulations. (https://github.com/mad-lab-fau/BioMAC-Sim-Toolbox). It is not necessary to have a background in human gait and biomechanics, as the analysis will focus on the algorithm performance.

The tasks are as follows:

- Literature search into available open-source gradient-based optimization algorithms and internal algorithm settings
- Selection of suitable algorithms and internal settings
- Creation of experimental procedure to compare algorithms
- Analysis of algorithm performance based on experiments

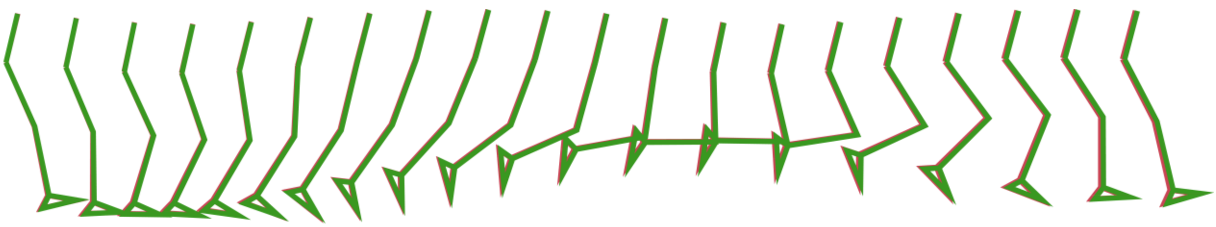

Figure 1 - Example running simulations visualized with stick figures