1 Software User's Guide

This file gives a user's guide accompanying the code that contains the software used to obtain the results of the accompanying thesis. It will exclude the OpenSim dataset and only show the three-dimensional case, since this is the default use-case. Known bugs and minor improvements to be done are tracked in `https://github.com/sachinumans/SAUmansThesis`.

2 Training

To train the model, open `System>Model>TrainModelAndFPE.m`. The first 62 lines have to be adjusted to fit the data structure of the gathered optical marker measurements. In these lines the optimisation weight for the model states are also defined, as well as the ground reaction force threshold used to detect heel strikes. The rest of this script should function without adjustments. Note that the script will run significantly faster if the `Parallel Computation Toolbox` in MATLAB is installed. The user will be prompted with the option to tune the initial leg parameters manually if desired, but this is often unnecessary. After the training procedure, the gait phase transitions are analysed and plotted to check there are no missing reset maps. On line 348, the value of ϕ_{sway} is entered as a function input. The trained parameters are saved on line 354.

The value of ϕ_{sway} can be determined by running `System>Model>OrientationEstimationTuner.m`. The start of this script needs to be adjusted to the data structure again. After running once, the frequency spectrum of the sway will be plotted and the value of ϕ_{sway} can be chosen. Then, on line 124 the determined ϕ_{sway} can be entered to check if the estimation works.

The base of support values can either be determined heuristically, or if measurements at multiple walking speeds are available, `System>SystemBlocks>XCoM>TrainXCoM.m` can be used to analyse the measurements. It is likely that the first 50 lines will need to be adjusted to the specific datastructure. The analysis itself, however, is performed on lines 55 through 95.

Before running any script containing an observer, the explicit equations of motion have to be updated by running `System>Model>SymbolicModelFunctions>Create_EoM.m`.

3 Running the Detector

The script `System>ExecutableScenarios>FallDetection>DetectFall.m` is currently configured to simulate a sensor failure due to the lack of falling data. If the user inputs falling data in lines 52 to 84, the lines 20, 21, 220-229, 343-345 should be removed. The filter parameters are defined in lines 30 to 50 and the sensor position is set on line 105. These should be the only adjustments necessary to run this script and the result will be plots of the estimated state, including detection time.