Ana de Sousa web: *linkedin.com/ana.sousa* anacsousa@lara.unb.br

FES gait (one-leg model) [2019-10]

About the package

In previous work, we created a simulation environment for FES gait control of one leg (de Sousa et al., 2019). The system tracks knee and hip trajectories of a hybrid neuroprosthesis (HNP), composed of a hip orthosis and FES-controlled knee motion. The newest version contains four different FES knee controllers. The control architecture (Figure 1) is composed of: (1) a torque proportional-integral-derivative (PID) controller that represents an active orthosis to generate hip movement, and (2) an FES (2a) bang-bang (BB), an FES (2b) PID, an FES (2c) PID iterative learning (PID-ILC), or an FES (2d) PID tuning using extremum seeking (PID-ES) controller.

The code was run with MATLAB R2018a, OpenSim 3.3, Windows 10 Home 64-bit Operating System, and uses the OpenSim API in Matlab[1].

OpenSim model: In the folder *yourpath\FES_Gait_2019\Model*, you find the FES one-leg gait model. We represent the hip orthoses as a torque at the hip joint. You can visualize the model loading *hip_torque100.osim* in OpenSim 3.3 (this step is not necessary to run the code). This file was created with the script *createModel.m* (lines 14 to 22).

Libraries: In the folder *yourpath\FES_Gait_2019\OpensimLibrary*, you find scripts from the OpenSim API. We do not recommend changing any of those files.

In the folder *yourpath\FES_Gait_2019\GaitLibrary*, you find specific scripts files for the gait simulations. Be careful when changing these scripts.

In the folder *yourpath\FES_Gait_2019\Dataset_hipsuit*, you find specific data from a subject walking at a slow-speed rate. The controllers use this data for hip and knee tracking control.

Run it: For running the code, it is necessary to describe the initial configuration. There are two examples of those configurations in files *yourpath\ FES_Gait_2019\RUNME_GAIT.m*. You may change the parameters in this file.

During the run, we may see some information, e.g., *6/635: knee: -26.524120, u_q: 0.000000, u_h: 0.000000, e: -1.433290*. That means that it is the 6th simulation of 635 simulations, the knee angle is at -26.524120º, the quadriceps excitation is 0, the hamstrings excitation is 0, and the knee error is -1.433290º.

The simulations may take a while. Depending on the computer, a 20 seconds simulation may take around 15 minutes.

When finished, Matlab saves the workspace data and opens a figure showing the plots.

Results from simulations: After each simulation, the code saves the results in *yourpath\ FES_Gait_2019\Results*. I added one example for better understanding:

- gait__Tf12.7049_CE_PID_CF_Open-loop_STIM0.0444_F0_R1_Kmax_s_r.sto
  After loading the model in OpenSim, you can load this .sto file and visualize the gait (this step is not necessary to run the code). If the path is too long, OpenSim may present an error, try transferring this file to another location, e.g., Desktop.

- gait__Tf12.7049_CE_PID_CF_Open-loop_STIM0.0444_F0_R1_Kmax_s_r.mat
  When we open this file in MATLAB, a figure and the workspace is loaded. The figure presents four plots over time. (1) the plot shows quadriceps, hamstrings and hip actuation. (2) reference and simulated knee angle. (3) reference and simulated hip angle. (4) PID parameters (they change only for the PID-ES).

R1_Kmax_s_r The files are named based on the initial parameters, i.e., final time (Tf5, 12.7049 seconds), hip active orthosis control (CE_PID, i.e., PID control), FES knee control (CF_Open-loop, i.e., BB control), excitation for muscles (STIM0.0444, i.e., muscles are excited at 0.0444 intensity if BB is on, goes from 0 to 1), fatigue (F0, i.e., no simulation of fatigue), which dataset to use (R1, i.e., dataset gait01, when the subject was walking at 0.1m/s), passive orthoses on the knee (not yet implemented in this version).

The script overwrites old files if the newest file contains the same basic configuration (usually because those files contain the same results). If you need to keep old data, it is necessary to transfer them to another folder.

[1] Follow the instructions for *Setting up your Matlab Scripting Environment*:
https://simtk-confluence.stanford.edu:8443/display/OpenSim33/Scripting+with+Matlab

Final notes: The RUNME files call for gait_script.m (*yourpath**FES_Gait_2019*), which initializes the system and calls for the OpenSim API along with the gait_control.m (*yourpath**FES_Gait_2019**GaitLibrary*). The gait_control.m reads the angles and, based on the initial setup, calls for the script corresponding to each controller (e.g., hip_EXO_pid.m and knee_FES_pid_ilc.m).

With the results of the chosen controller, gait_control.m applies excitation to the muscles, and the actuation to the hip.

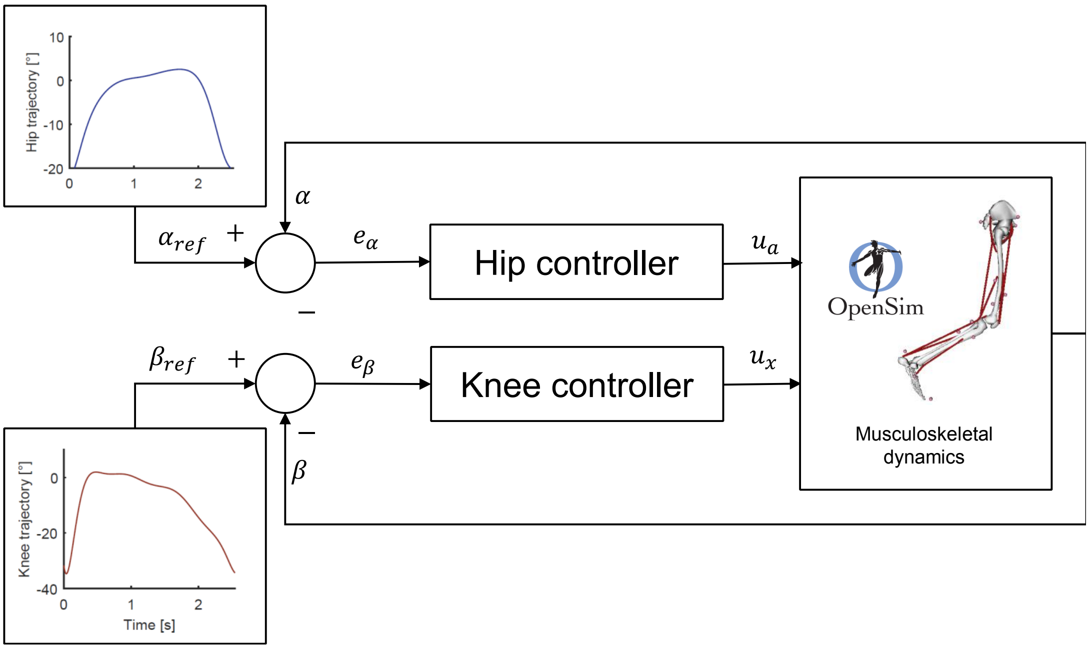

Figure 1: In this gait control architecture, the knee controller provides an excitation signal for the muscles u_x based on the error (e_β) between the reference knee joint angle β_{ref} and the measured angle β. The hip controller provides a signal u_a based on the error (e_α) between the reference hip joint angle α_{ref} and the measured angle α. Both controllers consider the predefined hip and knee trajectories. With the control signals (muscles excitation and torque), OpenSim calculates the musculoskeletal dynamics.

References

de Sousa, A. C. C., Freire, J. P. C. D., and Bo, A. P. L. (2019). Integrating hip exosuit and FES for lower limb rehabilitation in a simulation environment. *IFAC-PapersOnLine*, 51(34):302–307.