rameter, higher will be the generalization encompassed by CMAC.

ANNs as machine learning algorithms must be trained. Learning can be supervised or not supervised. CMAC's case is supervised learning. During this phase input signals and the desired output signal must be given for CMAC's learning algorithm. The last parameter to inform is the iterations number of the CMAC. This parameter is the number of times the training algorithm will update the weights after calculating the output for all input signals previously collected.

D. Simulation and implementation details

For CMAC's implementation and knee's simulation was adopted the Python 2.7 programming language, the software packages NumPy and Matplotlib were used too. The operating system is Mac OS X Yosemite version 10.10.1. The hardware is a Mac Pro (Mid 2010), processor 2.8 GHz Quad-Core Intel Xeon, RAM 3GB. In addition, the source code to the created simulations is available in http://github.com/rob-nn/motus.

As input signals for the simulation, velocities vectors in a 3D plane (x, y, z) of the left knee were chosen. As output, was chosen to predict the signal corresponding to the right knee angular velocities. These choices were made based on work [12].

Before running the simulation, the RNA CMAC must be trained. For accomplish this task only 50% of the data collected in a single walk were used. The other 50% were used to make the prediction of the output signal and compare them with the real signal captured.

III. RESULTSS

Figure 4 shows the simulation performed. For this simulation was informed: 50 iterations, 30 activations and 200 discrete values per input signal. The input signals are left knee angular velocities in a 3D plane (x, y, z). The data are 50% of a 5 seconds walk. Parts of the first and last moments were discarded. The output is the angular velocities of the right knee.

In addition to the generated simulation, an open source project, called Motus was created. See Figure 5 This project aims to create a tool for basic gait analysis and signal simulation of human lower limbs. Figure 5 shows the version 0.1 of this tool, which is available on GitHub site at http://github.com/rob-nn/motus.py address. The available version is capable, a while, to work with the following signals:

1. Knee angles (flexion, extension);
2. Knee angular velocities (flexion, extension);
3. Knee angular accelerations (flexion, extension);
4. Knee velocities in a 3D plane.

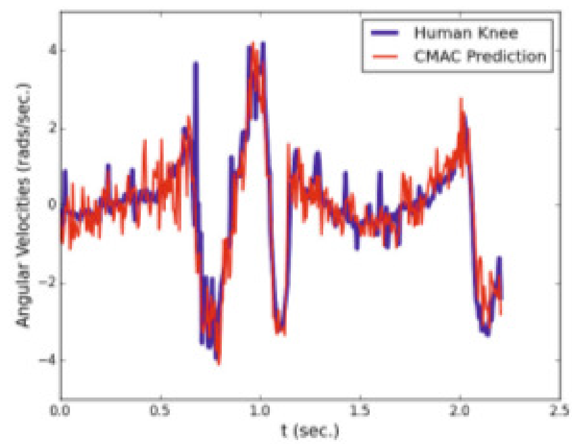

Fig. 4: Human knee angular velocity prediction

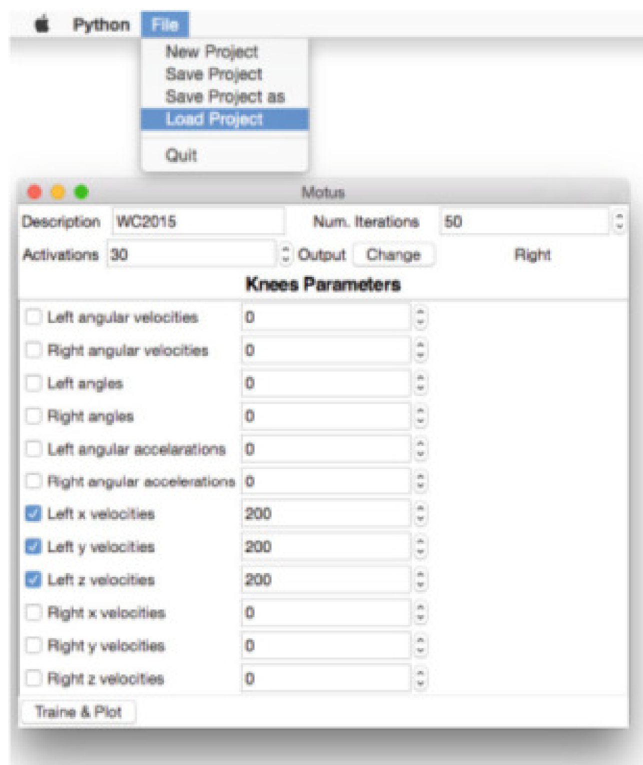

Fig. 5: Motus Version 0.1

Motus development is powered by Healthy Informatics Lab (*Laboratório de Informática em Saúde* – LIS).