

# Sequence Labeling for Gait Analysis using LSTM

Pablo A. Iturralde  
Bioengineering Department  
University of Pittsburgh  
Pittsburgh, PA  
pai7@pitt.edu

Yin Zhong  
The Robotics Institute  
Carnegie Mellon University  
Pittsburgh, PA  
yinzhong@andrew.cmu.edu

Jakob Bauer  
School of Computer Science  
Carnegie Mellon University  
Pittsburgh, PA  
jsbauer@andrew.cmu.edu

## 1. DATA

For the initial training purposes, we selected data from 8 healthy subjects walking in a treadmill at three different speeds for 100 seconds each. Additionally, we took data from 13 stroke subjects, walking at a single speed for 100 seconds. For each subject, the input data is 3D marker position of 18 different markers placed in the same anatomical positions for each subject, sampled at 100Hz. The event labels (target) were computed from ground force reaction data samples at 1kHz for healthy subjects, and 2kHz for stroke subjects. This is currently considered the gold standard [miller\_gait\_2009, miller\_gait\_2009, Hreljac2000].

## 2. ALGORITHMS

In order to compare our results, we implemented two algorithms presented in the literature: a heuristic-based method [oconnor\_automatic\_2007], and a neural-network approach [miller\_gait\_2009].

### 2.1 O'Connor's Heuristic

[oconnor\_automatic\_2007] describe a heuristic to compute gait events by just considering vertical and horizontal speed of heel and toe markers respectively, and using a peak detection algorithm.

### 2.2 Miller's Neural Network

[miller\_gait\_2009] is one of the few data-driven efforts of event estimation. It uses a classical NN to classify each sample. The inputs are computed from the position, instantaneous velocity and instantaneous acceleration of each available motion-capture marker in a time-window centered around the desired time sample. Because the inputs are too many (the total input size grows linearly with the window size and the number of markers), a dimensionality reduction technique (PCA) is used before training the NN.

## 2.3 Our approach: Long Short Term Memory Neural Network

Gait event detection is a form of *sequence labeling*, i.e., assigning a sequence of labels to a sequence of data. One approach to sequence labeling is the use of recurrent neural networks (RNN). However, due to the vanishing gradient problem RNNs cannot store information for a long time which means that their usefulness for sequence labeling is limited. One solution to this problem is the use of a Long Short-Term Memory (LSTM) architecture instead of conventional RNNs [graves\_supervised\_2012]. In LSTM, the summation units in the hidden layer are replaced with so-called memory blocks. These blocks consist of a memory cell as well as three multiplicative units called input gate, output gate and forget gate. The gates allow information to be stored in the memory cell for long periods of time. While we haven't decided on the overall design of our LSTM network yet, it will probably have specifications along the lines of:

- Up to 54 input units that each take the sequence corresponding to one marker as input; the reason we might not need all 54 marker sequences is that some of the markers (e.g., for the hip) presumably carry little useful information.
- A hidden layer of LSTM memory blocks; the exact number will depend on the number of input units.
- 2 output units (one for each leg); the output will be thresholded with '0' corresponding to stance phase and '1' corresponding to swing phase.

## 3. COMPUTATIONAL RESOURCES

For the LSTM network we are using an Amazon EC2 instance of type g2.2xlarge. This instance type comes with an 8 core 2.6 GHz Intel Xeon E5-2670 processor and an Nvidia GRID K520 GPU that is composed of two 4 GB GK104 GPUs. We chose Torch as our computing framework.

## 4. INTERMEDIATE RESULTS

We implemented and tested the two methods described in [oconnor\_automatic\_2007] and [miller\_gait\_2009]. Results are in line with the ones presented on each paper. Summaries of the true error (detected event time minus actual event time) can be found in Figures 1 and 2. There are no results yet for the LSTM approach.

## 5. DIFFICULTIES AND FUTURE WORK

In the following weeks we will be implementing an LSTM network on Torch. As we have no experience using Torch in the past, we are not sure how long it will take.

Other future work includes feature engineering to determine the most relevant input variables, from all the available marker position data and its derivatives, in order to reduce the size of the network that has to be trained. We would also like to extend the capabilities of the method to data that was acquired in natural walking (as opposed to treadmill walking). We believe adequate feature engineering may be enough to achieve this objective too.

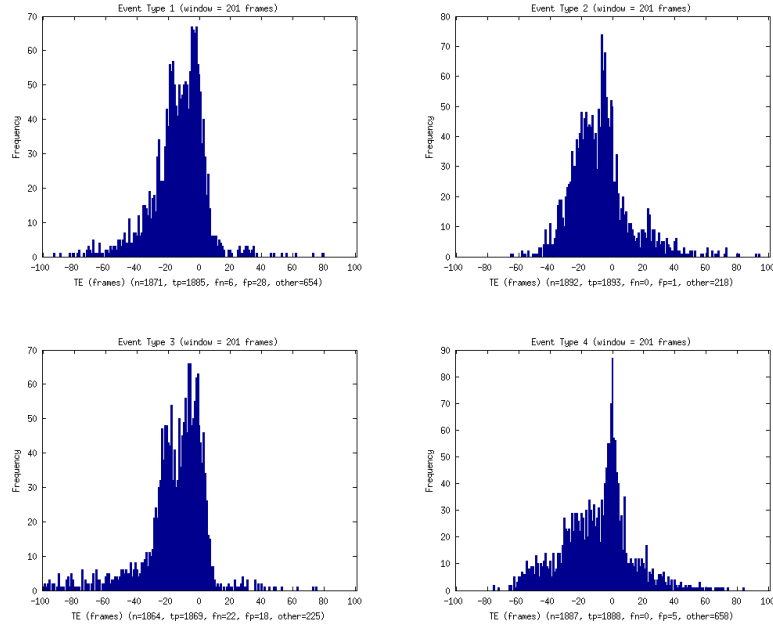


Figure 1: True errors per event type for O'Connor's heuristic. The data set consists of 8 subjects à 3 trials each. The panes are (clockwise starting from the top-left): Left Heel Strike, Right Toe Off, Right Heel Strike, Left Toe Off. One frame corresponds to 1 ms. If the estimated event was not within 100 frames from the true event then it is reported as "other" and not shown in the histogram.

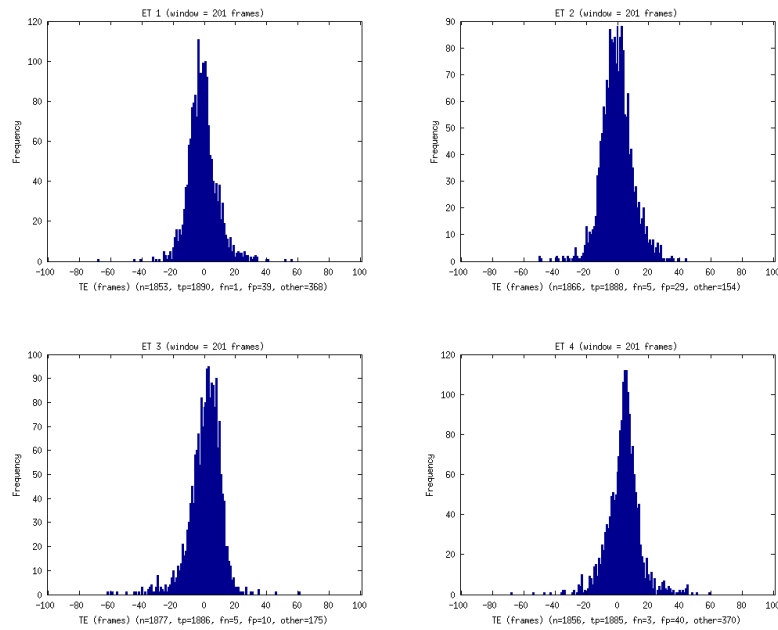


Figure 2: True errors per event type for Miller's neural network. The data set consists of 8 subjects à 3 trials each. The panes are (clockwise starting from the top-left): Left Heel Strike, Right Toe Off, Right Heel Strike, Left Toe Off. One frame corresponds to 0.5 ms. If the estimated event was not within 100 frames from the true event then it is reported as "other" and not shown in the histogram.