Stride Prediction for Exoskeleton Control

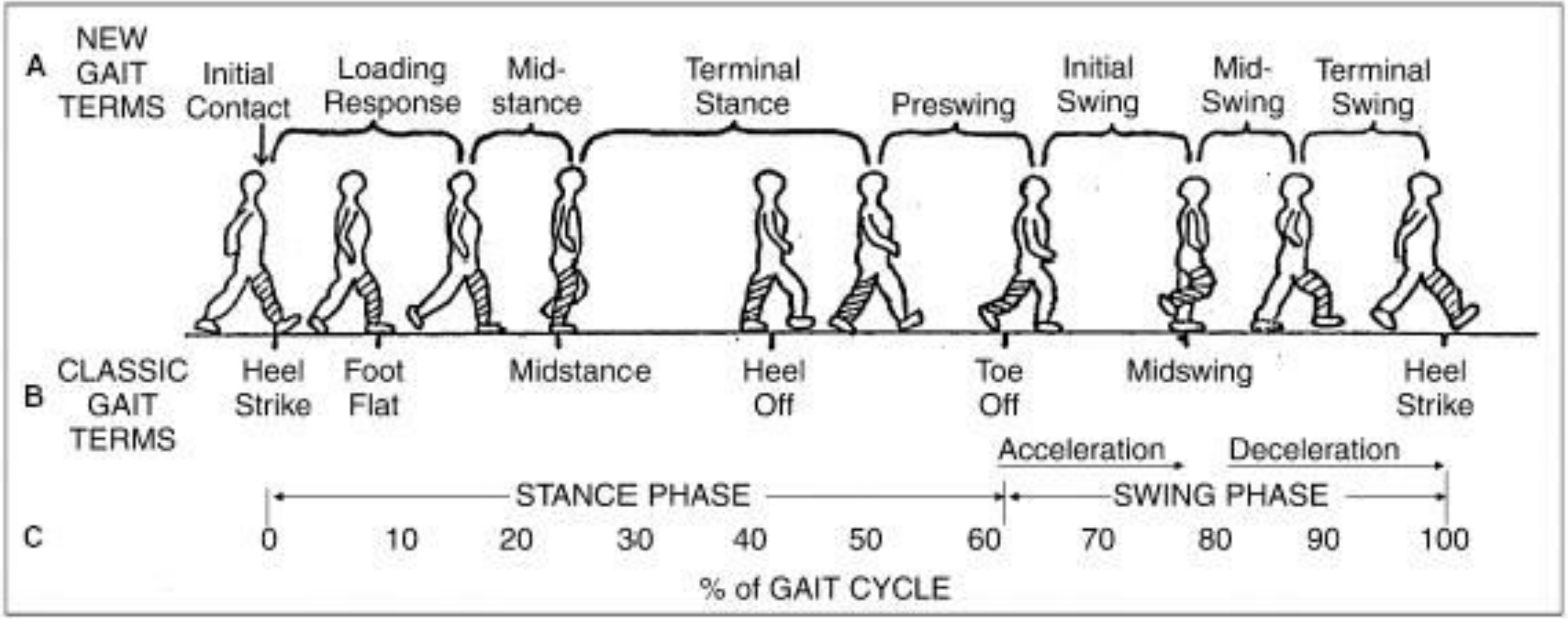
Jennifer Dawkins and Brian Do

December 12, 2018

Ankle or calf-based exoskeletons can provide assistance to those with neurological deficits or improve walking efficiency

…yet powered exoskeletons require stride prediction to provide accurate control

The components of the gait cycle are predictable



Our Data

6 DOF Accelerometer data for various walking instances over 15 minutes (~600 steps): mostly upstairs, downstairs, level ground, transition steps

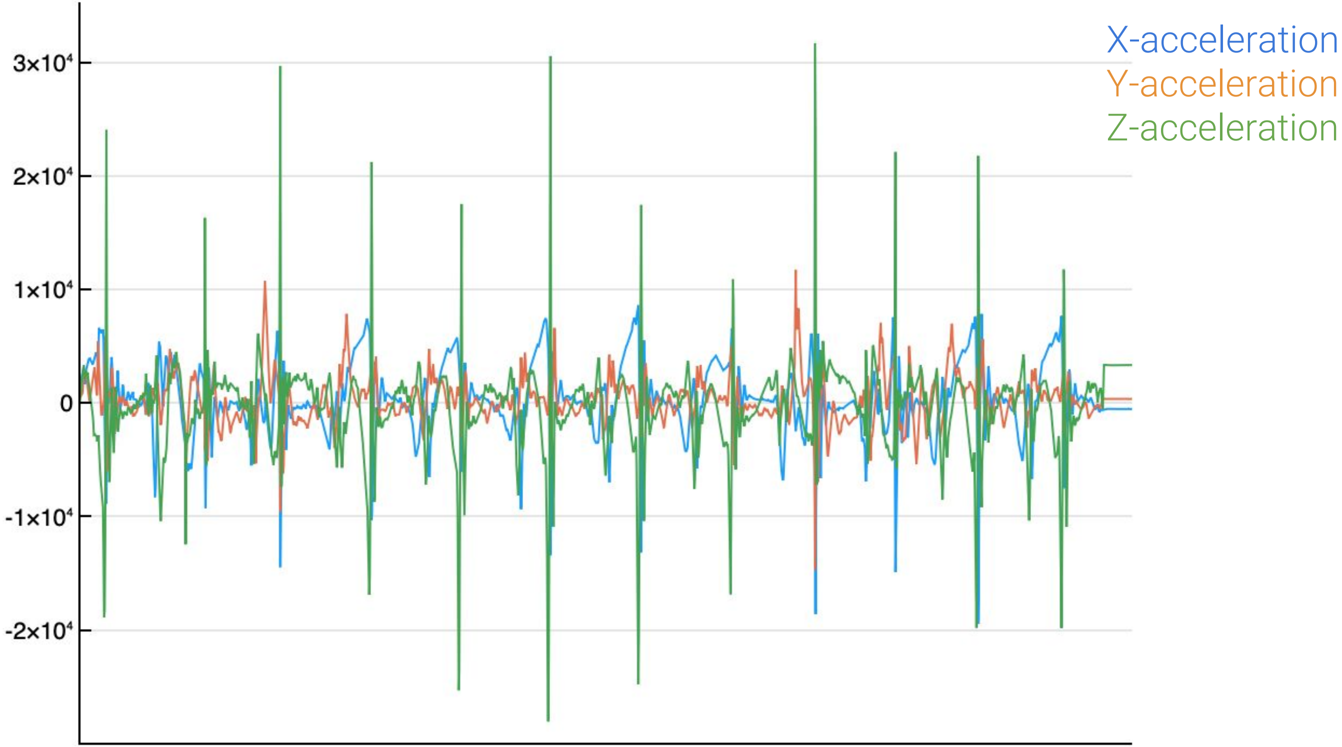
Powered exoskeleton control must adapt to different people and terrains. How to achieve this control?

Possible solutions:

1. Train a model to predict terrain
   a. Cluster based on data observed to date... but inflexible
2. Train a model to predict force
   a. Based on F = m*a
3. Train a model to ‘learn’ the shape of a stride
   a. Requires gathering data to correlate stride to force

We focused on 2 & 3

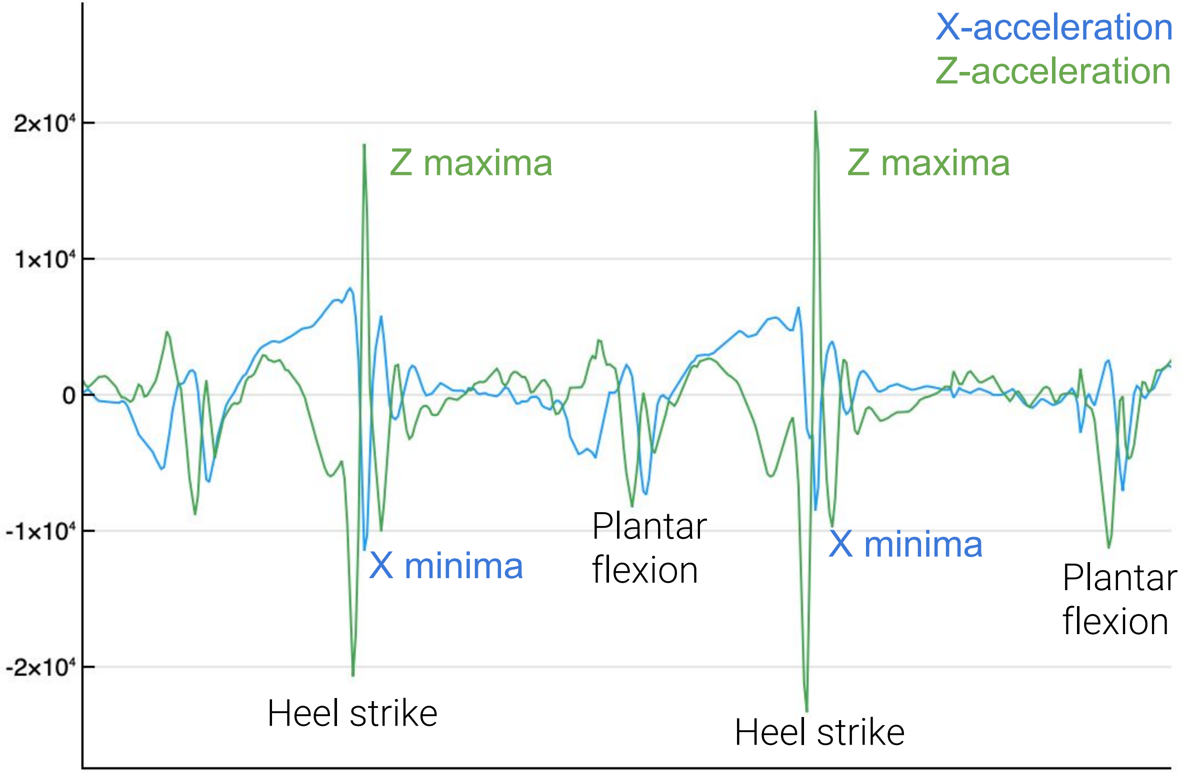
What does the signal look like?



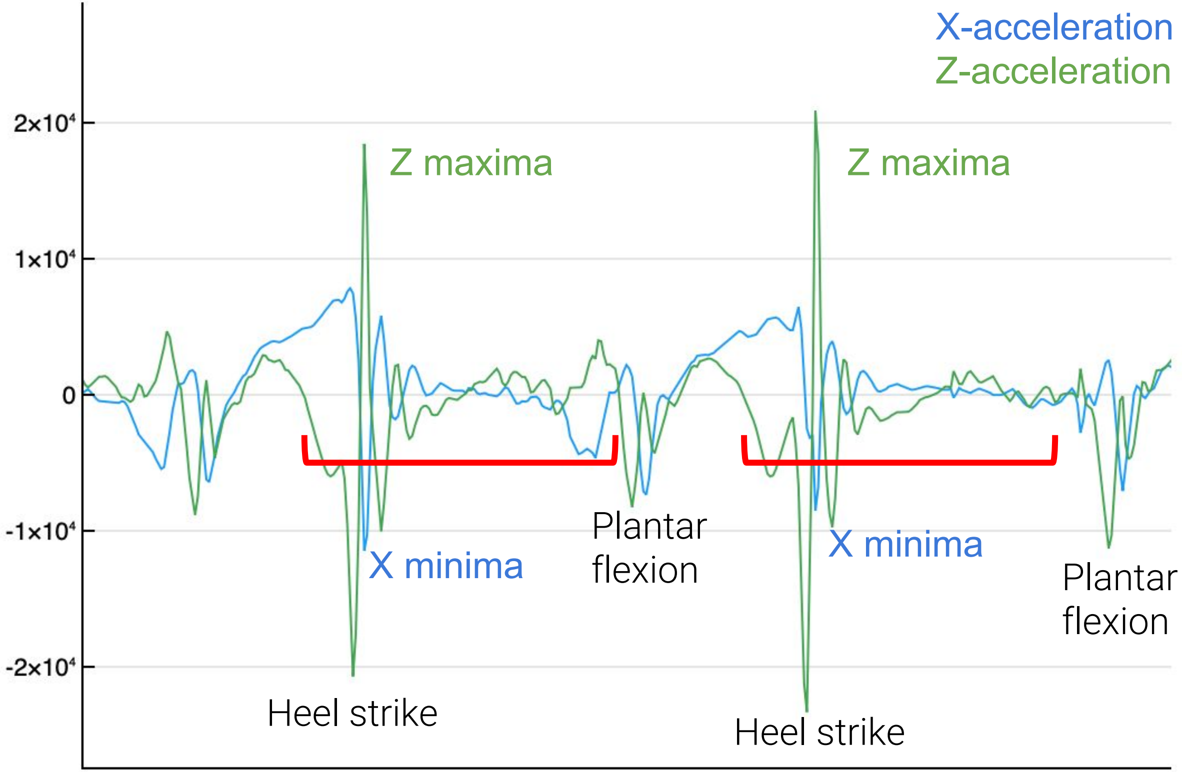
How do we annotate the signal in real time?

How do we annotate the signal in real time?

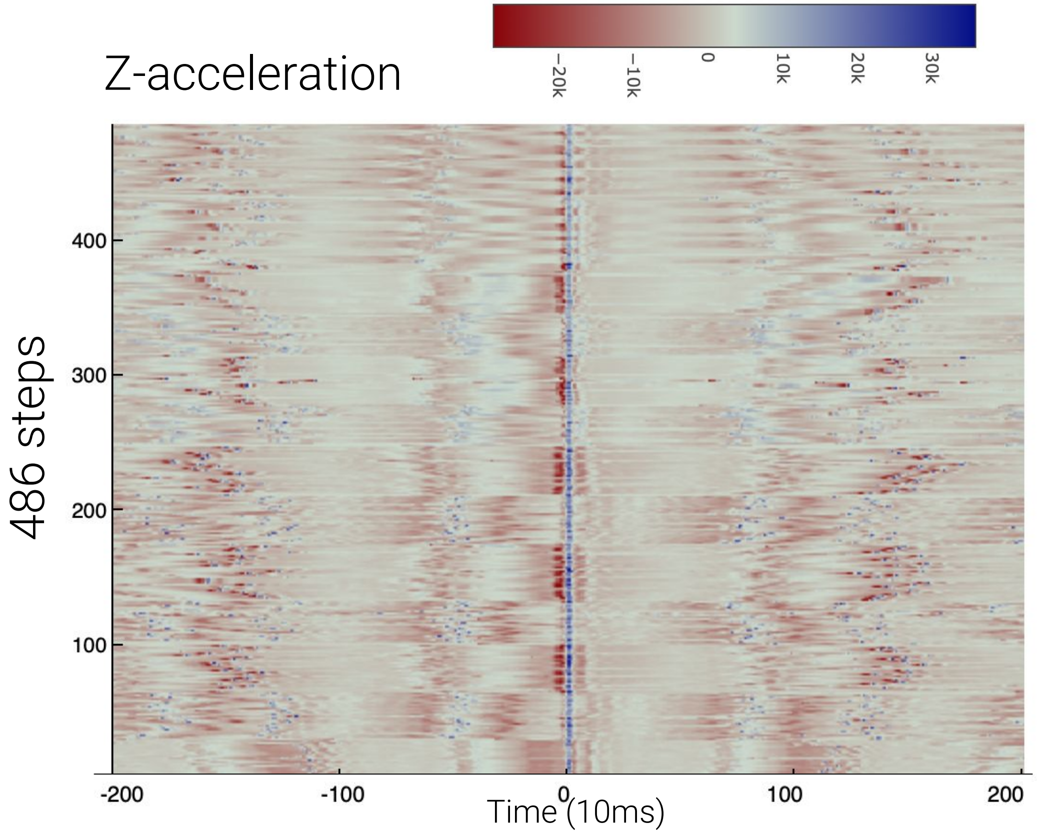
How do we annotate the signal in real time?



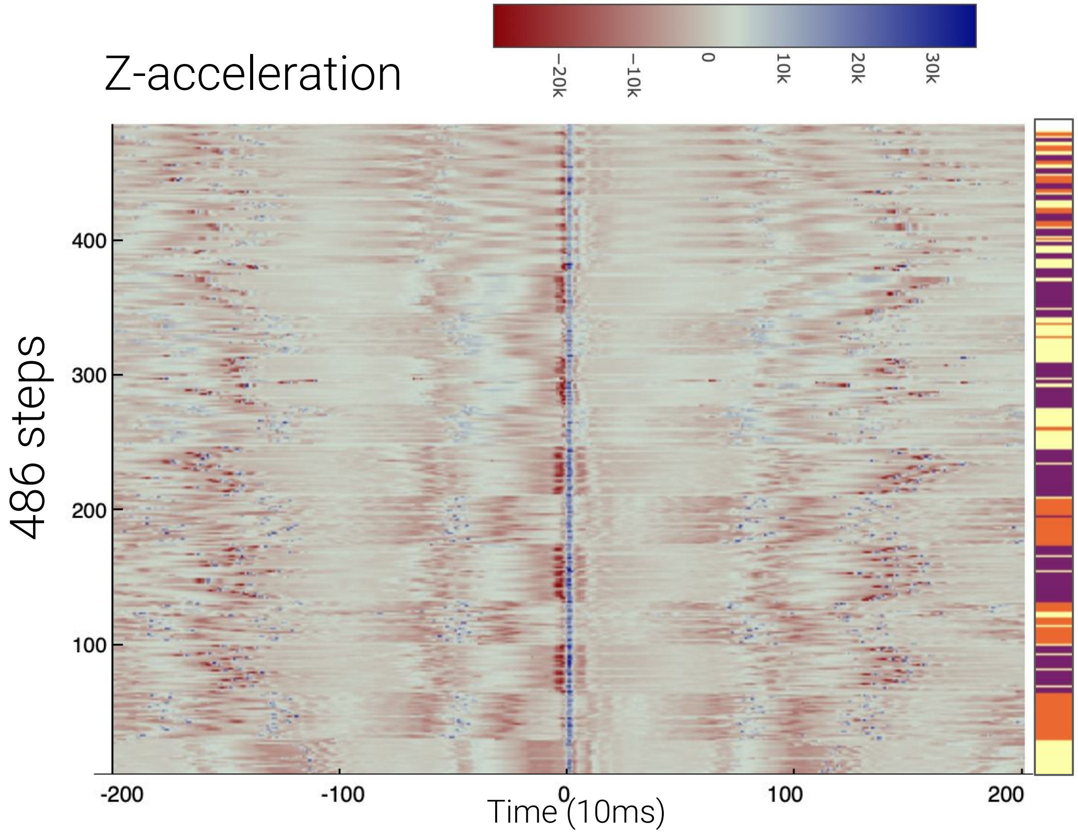
There is enough of a delay after heel strike to enable prediction of plantar flexion force



Then what do “typical” steps look like?

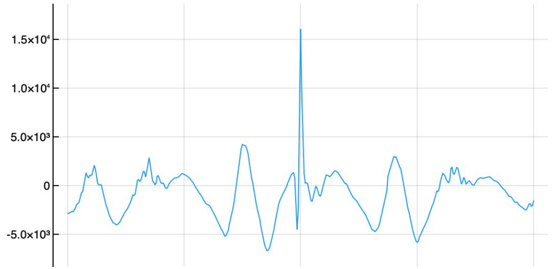


Then what do “typical” steps look like?



Cluster 1

Cluster 2

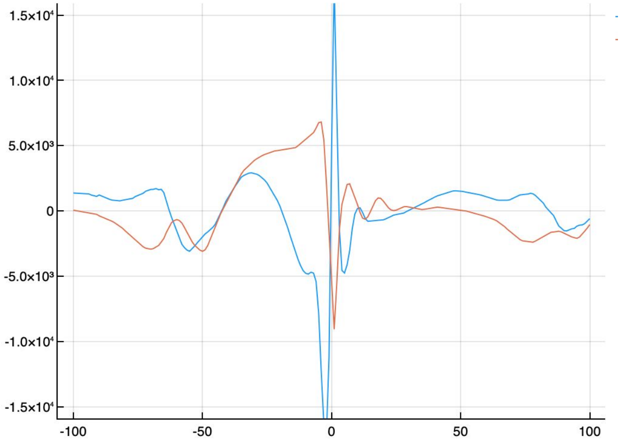


Cluster 3

Clusters

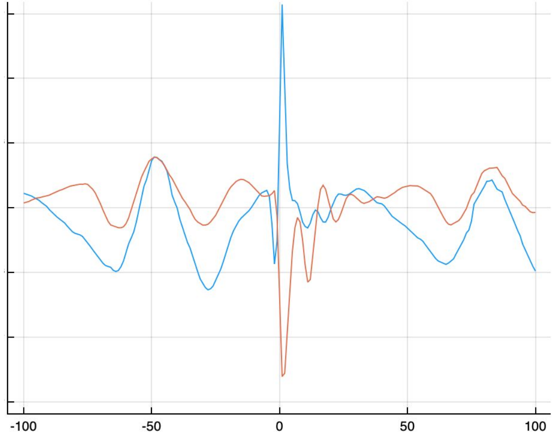
X-acceleration

Z-acceleration



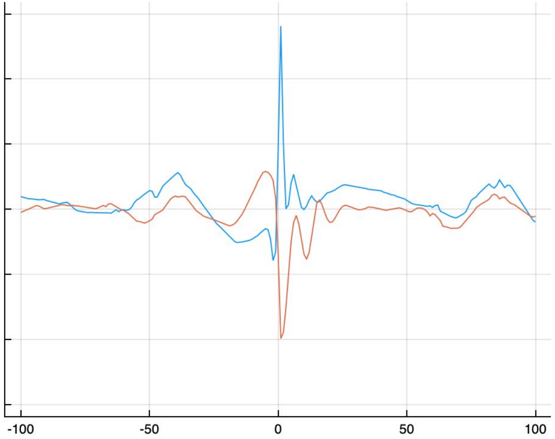
Cluster 1

Downstairs



Cluster 2

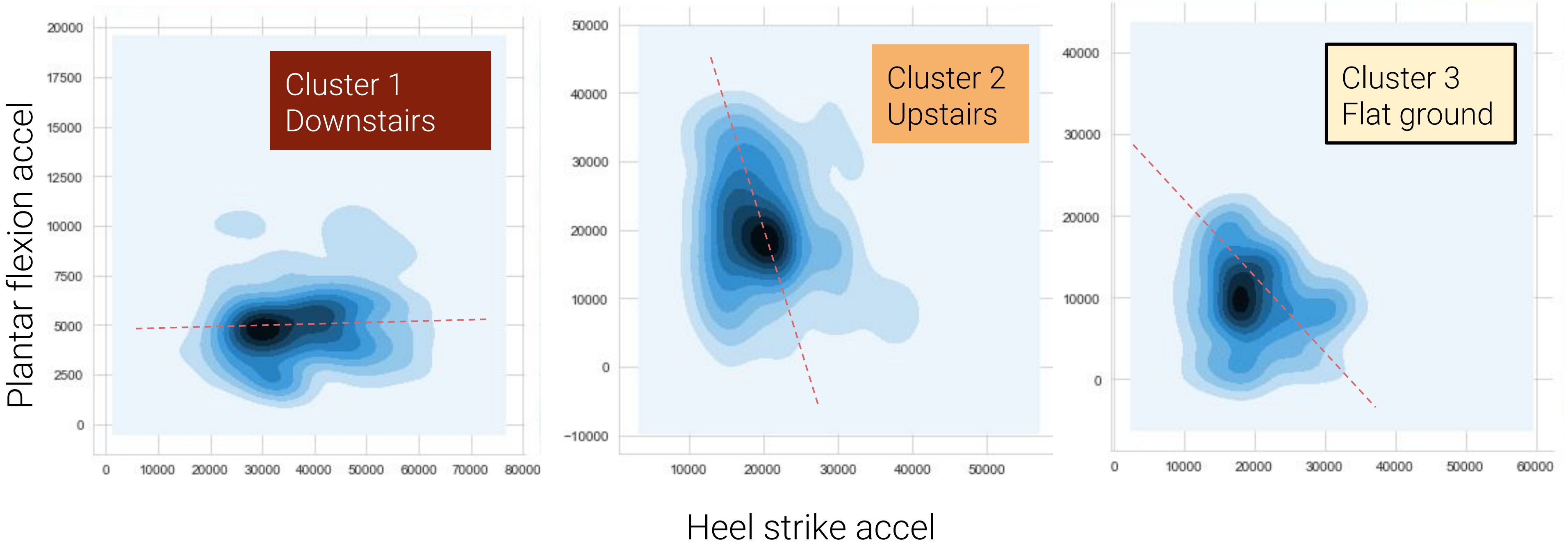
Upstairs



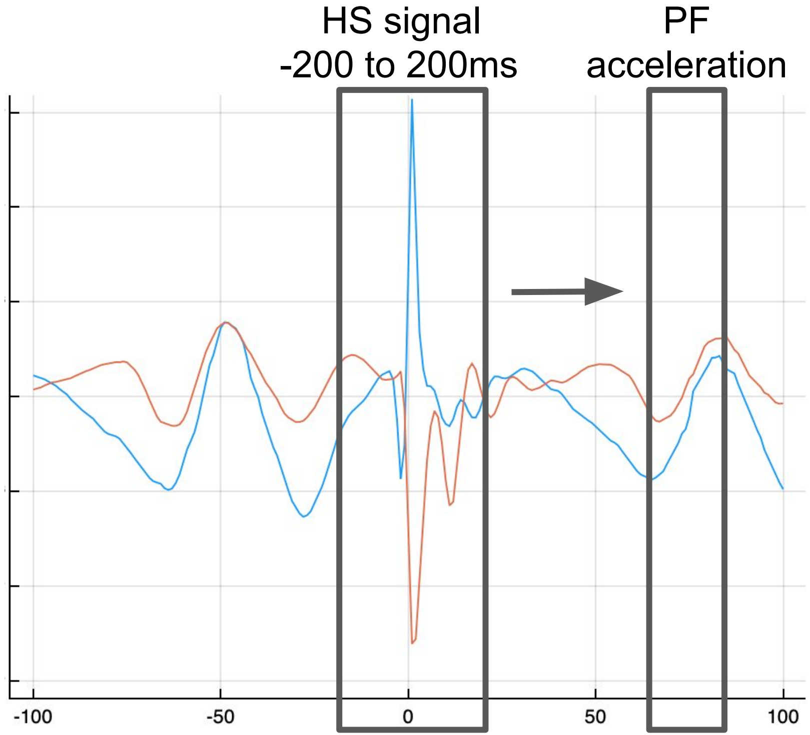
Cluster 3

Flat ground

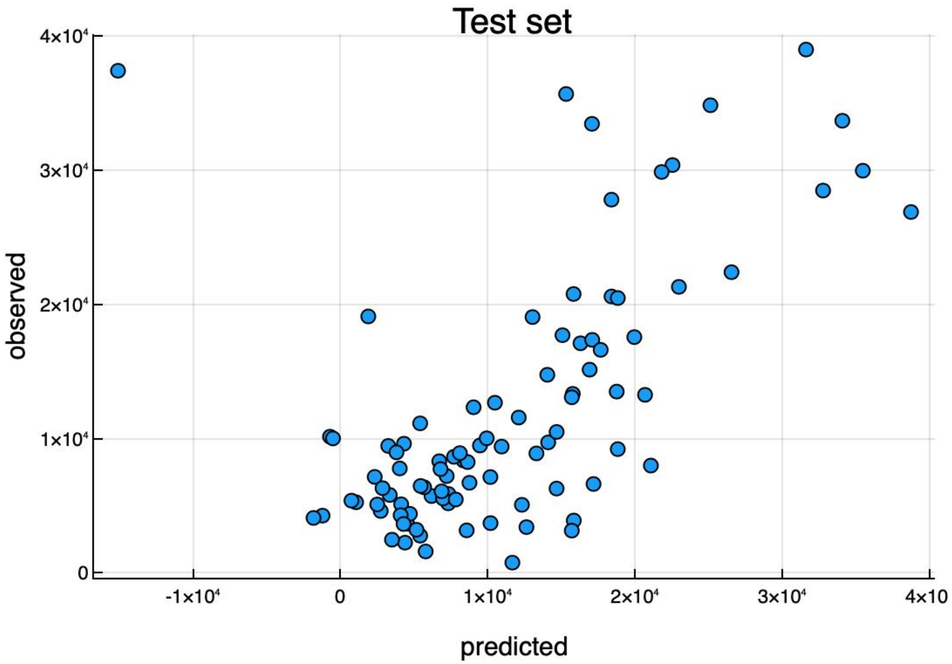
Plantar flexion acceleration associates differently with heel strike acceleration in different clusters



Can we learn PF acceleration from the shape of the heel stride, not just the magnitude?



Prediction of PF z-acceleration using a linear model with HS x and z acceleration achieves $r^2 = 0.6$



All of the above assumes that you can apply a static force to cause plantar flexion, but this is likely to cause a pretty inflexible gait.

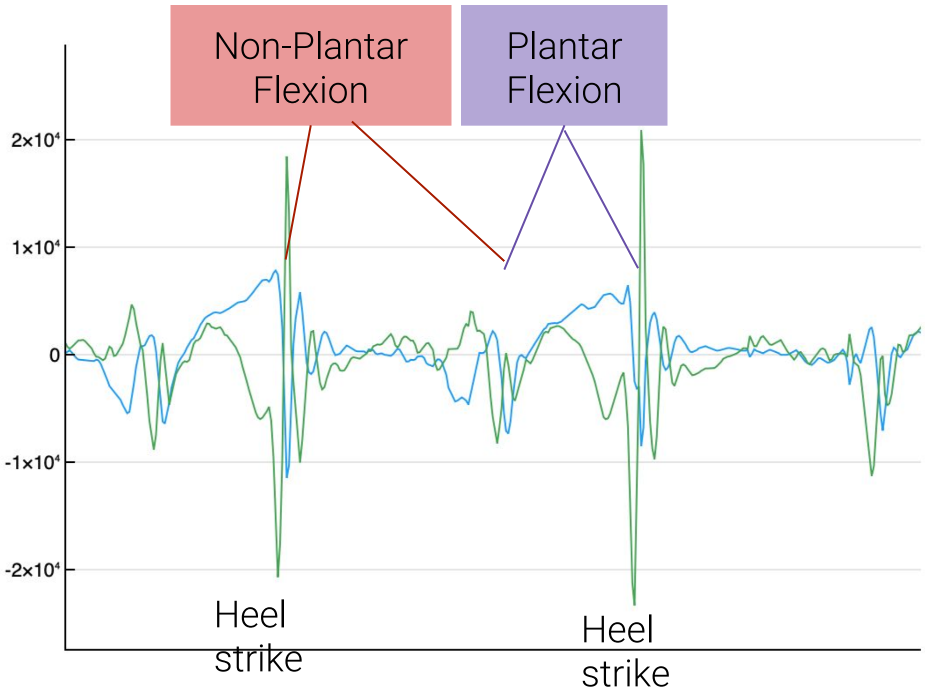
It would be better to learn the actual PF signal.

We also need to know the **timing** of the PF signal, not just the force to be applied.

Learning Strides

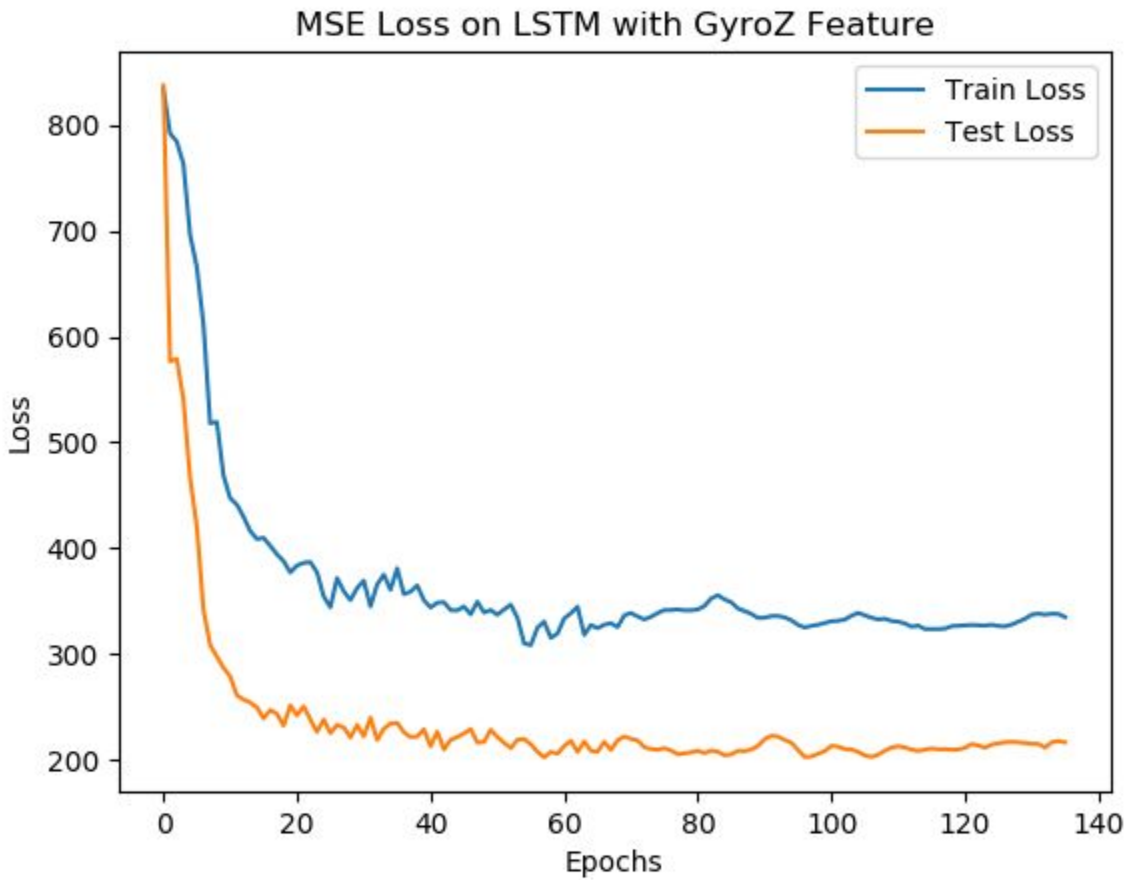
Adapt LSTM language learning models to stride learning models

- Method 1: Standard LSTM
  - Predict "plantar flexion" portion of stride from non-plantar flexion portion of stride by simply learning structure of each stride
- Method 2: Seq2Seq Model
  - Learn plantar flexion sequence from non-plantar flexion sequence



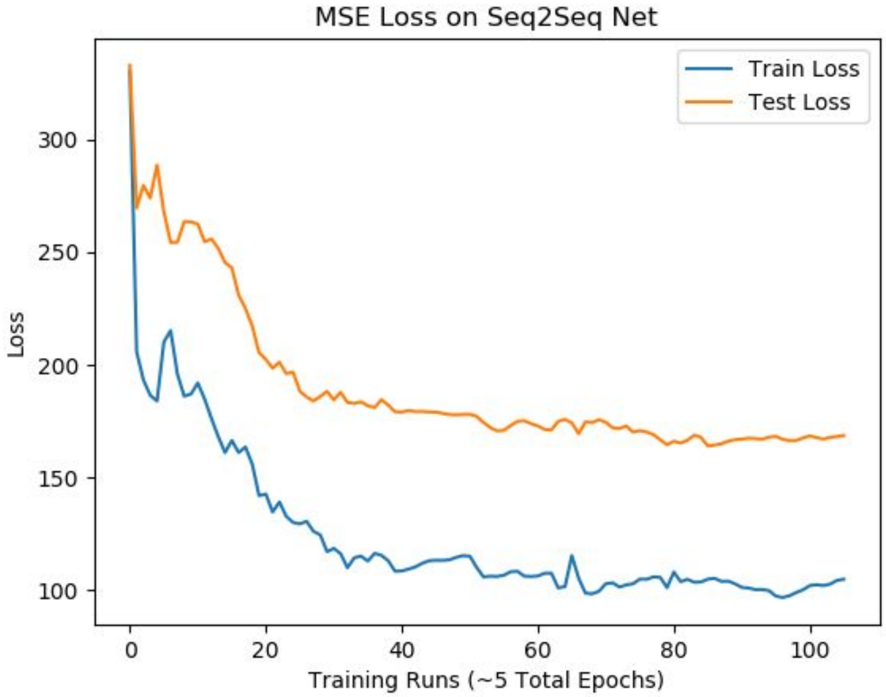
Both utilize PCA to reduce data dimensionality

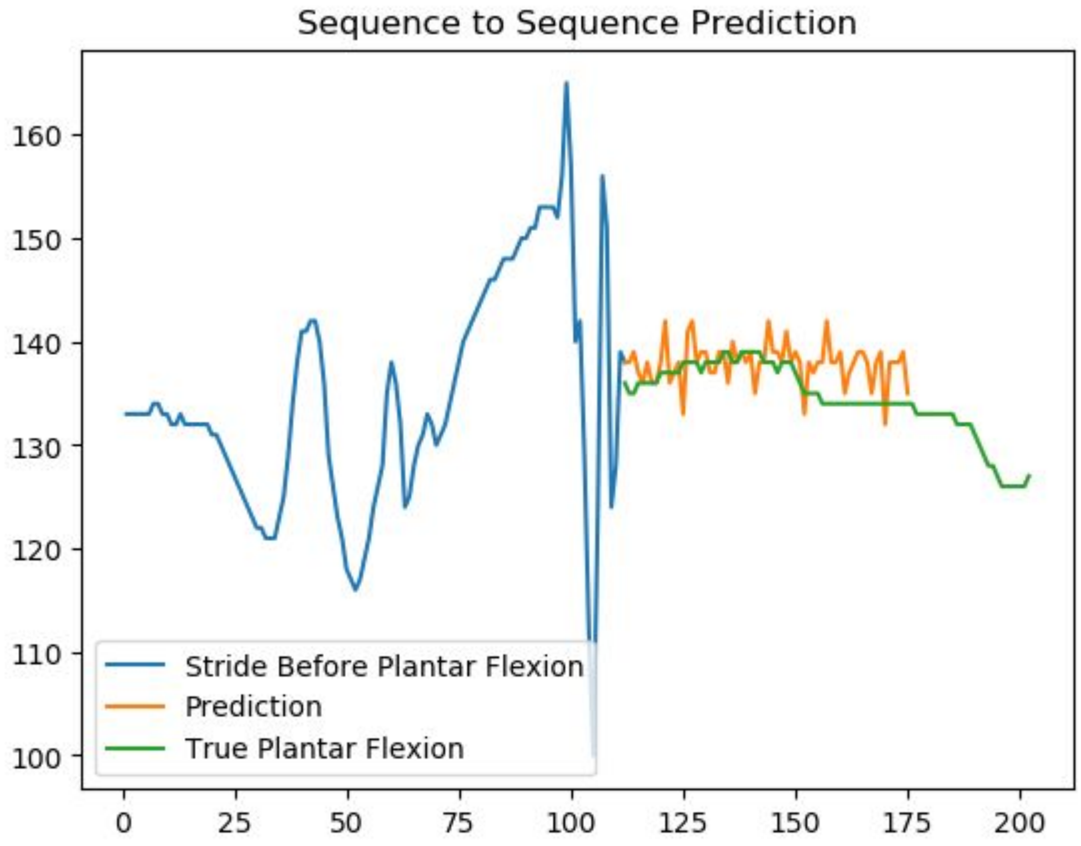
Results: Learn Stride



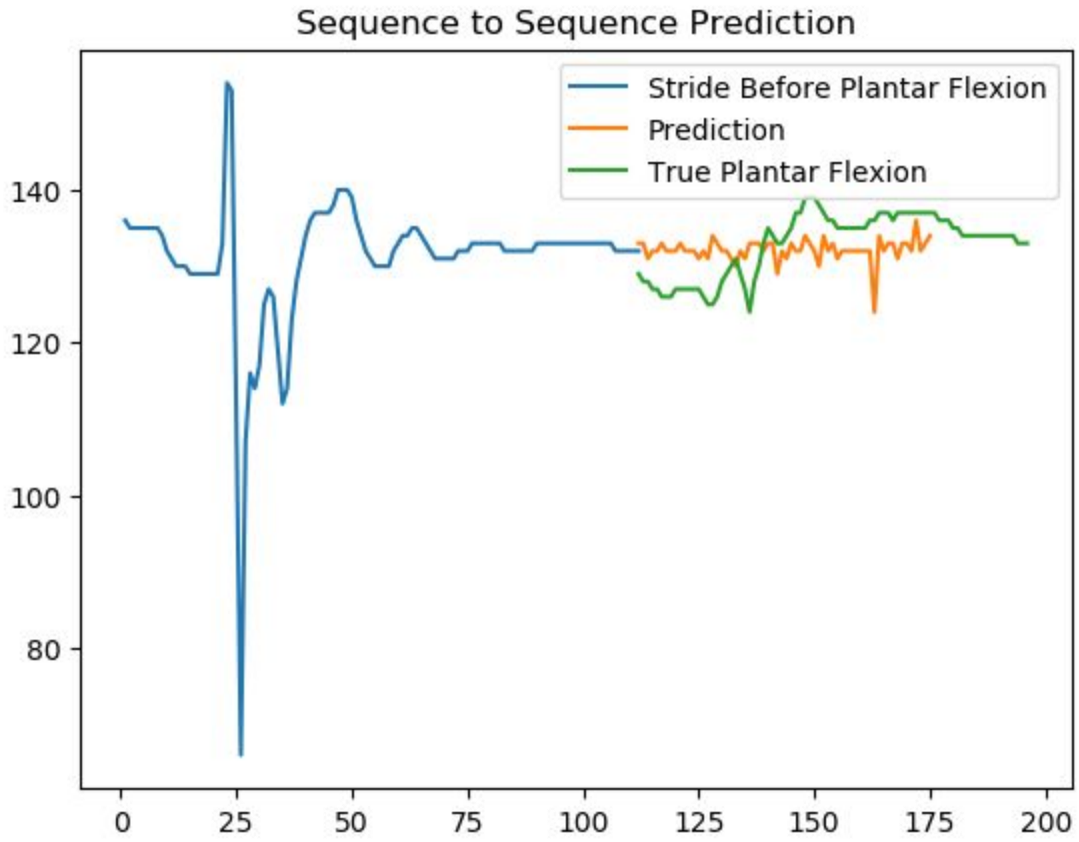
DEMO!

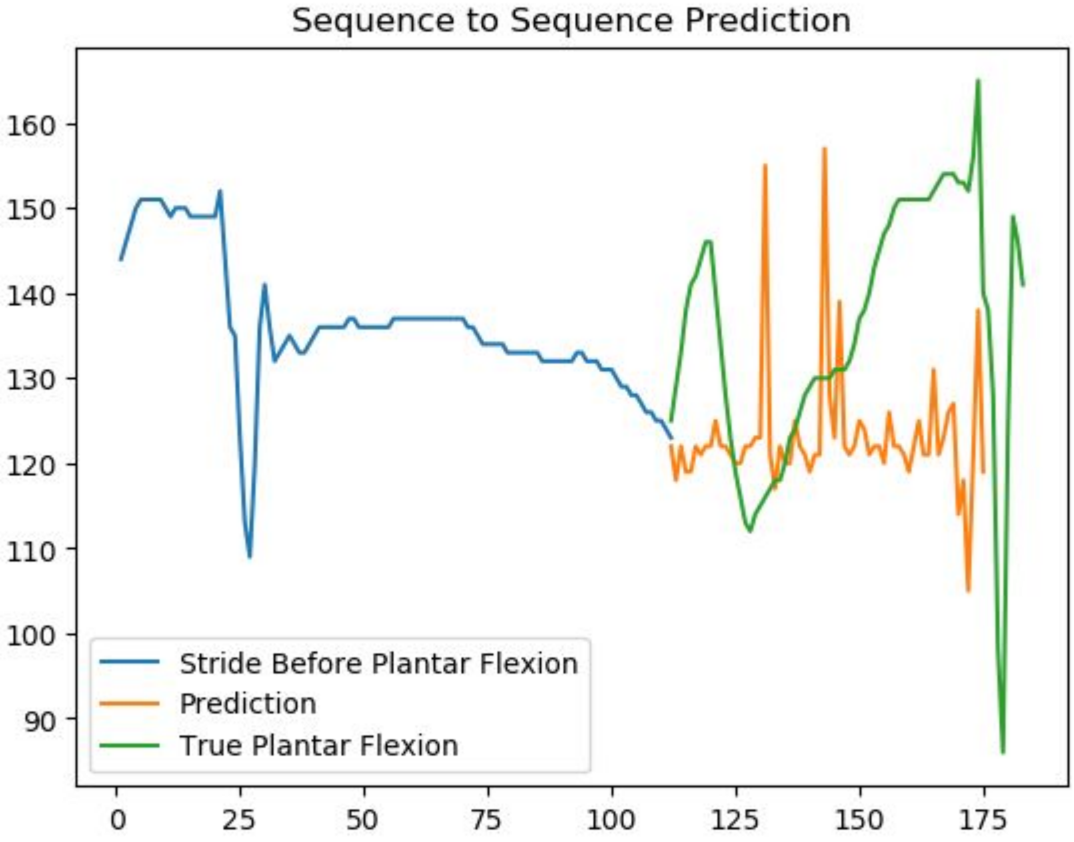
Results: Learn Plantar Flexion Sequence from preceding sequence





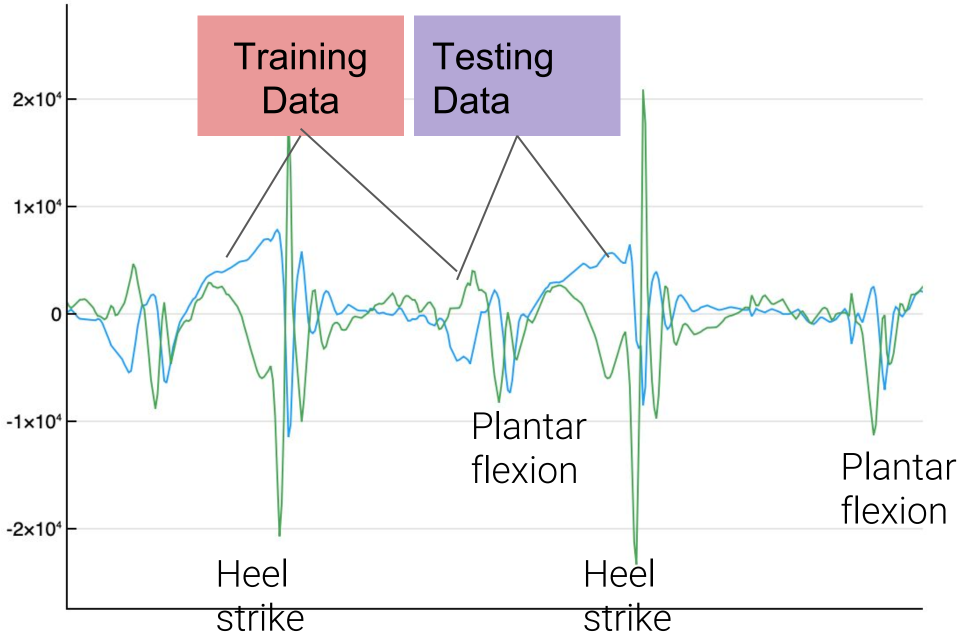
Sequence to Sequence Prediction
160
150
140
130
120
110
100
0
25
50
75
100
125
150
175
200
Stride Before Plantar Flexion
Prediction
True Plantar Flexion

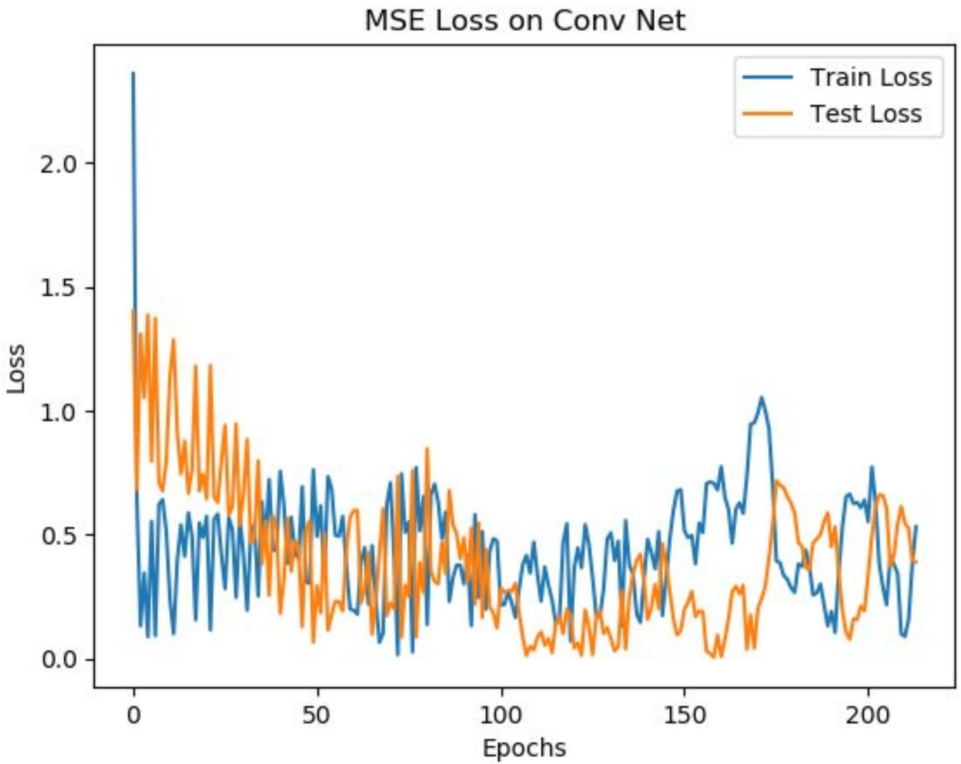

Sequence to Sequence Prediction
Stride Before Plantar Flexion
Prediction
True Plantar Flexion
140
120
100
80
0
25
50
75
100
125
150
175
200


Sequence to Sequence Prediction
160
150
140
130
120
110
100
90
0
25
50
75
100
125
150
175
Stride Before Plantar Flexion
Prediction
True Plantar Flexion

Predicting Forces

1. Calculate relative force as F = sqrt(ax^2 + az^2) of plantar flexion signal
2. Apply convolutional neural network to training data to predict this force for the coming plantar flexion




MSE Loss on Conv Net
Train Loss
Test Loss
Loss
2.0
1.5
1.0
0.5
0.0
0
50
100
150
200
Epochs

DEMO!

Conclusions

No method worked great, but most showed some promise

- Ability to learn something about strides

More exploration and optimization would definitely increase accuracy of results

- More epochs run
- Better data pre-processing