# Hand Activity Recognition
# Project Planning
# UW Madison - HCI Lab

Ruisen (Eric) Liu

May 11, 2017

# Contents

# 1 Essential Topics to Discuss

1. Therbligs - which ones need to be identified, and which ones identified by Dynamic Time Warping (DTW)?
2. The current training methodology is unreliable because it depends on a distinct person's measurements, and a specific orientation/reading from the armband. I think the input into DTW should be preprocessed to a standard norm.
3. Do we need to rectify/integrate the raw EMP data?
4. Current training registers for discrete test samples. How do we identify the right time series to feed into real-time application of DTW?
5. Hand Activity Level - Where does this fit in the big picture?

# 2 Objective

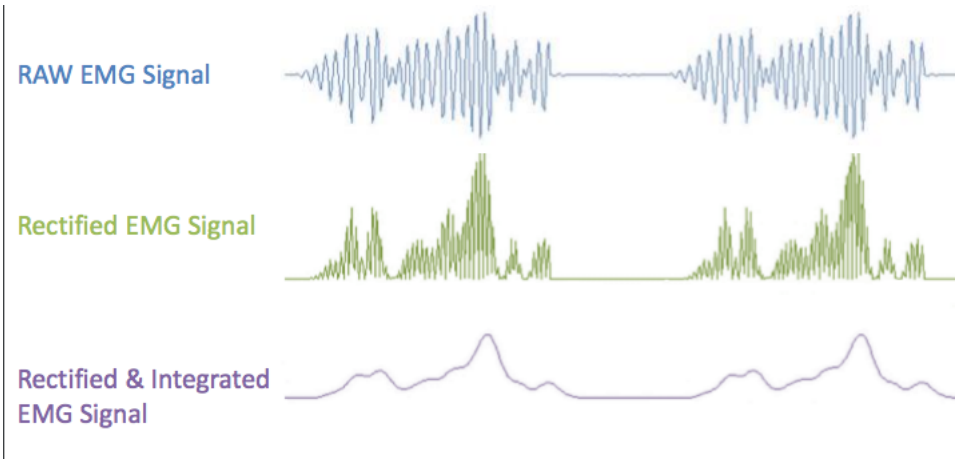
The objective of this project is to provide real-time therblig and hand activity level recognition through the use of a Myo Armband.

# 3 Hardware

The Myo Armband has a three-axis gyroscope, three-axis accelerometer, three-axis magnetometer, and 8 separate EMG sensors. The full tech specs can be found here.

Each EMG sensor measures muscle activation via electric potential. I suspect they are a built-in version of the Myo Muscle Sensor, which has the following specs found here.

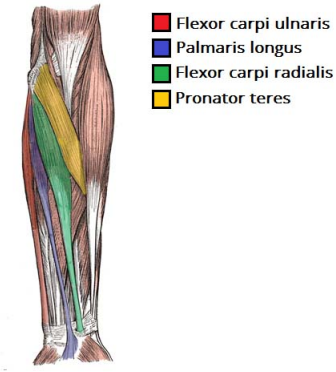
Thalmic Labs' SDK gives a pre-classified output pertaining to the motion type. However, they showed in a blog how to get the raw data from the EMG sensors. Unlike the Myo Muscle Sensor, there is no access to a Rectified & Integrated EMG Signal, such as below:

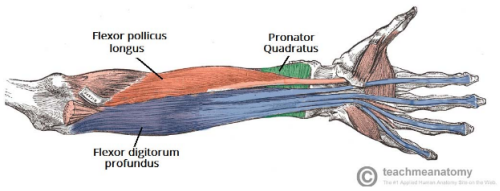



Therefore, we will either have to work with the raw data, or we may need to do rectification and integration in live-time on our own.

# 4 Background

1. Reference pictures for forearm muscles:




   Given the orientation of the EMG sensors, they will probably react well to flexion. These diagrams will be useful for calibration purposes.

2. **N-Dimensional Dynamic Time Warping**

   Dynamic Time Warping is a method by which to compare two time series with temporal differences. The following derivation is for N-Dimensional vectors in the time series.

   This is adapted from Gillian et al.

   Let there be two N-dimensional time series,

$$\mathbf{x} = \{\vec{x}_1, \vec{x}_2, ..\vec{x}_{|\mathbf{x}|}\}^T \; and \; \mathbf{y} = \{\vec{y}_1, \vec{y}_2, ..\vec{y}_{|\mathbf{y}|}\}^T$$

,
where each vector is of size N.

We look to construct a warping path $\mathbf{w} = \{\vec{w}_1, \vec{w}_2, ..\vec{w}_{|\mathbf{w}|}\}^T$ where the kth value of $\mathbf{w}$

$$\vec{w}_k = (\vec{x}_i, \vec{y}_j)$$

is an $(\vec{x}, \vec{y})$ pair chosen from their respective time series. The length, $|\mathbf{w}|$ is bounded by:

$$max\{|\mathbf{x}|, |\mathbf{y}|\} \leq |\mathbf{w}| \leq |\mathbf{x}| + |\mathbf{y}|$$

In addition he warping path must adhere to the following constraints:

(a) $w_1 = (1, 1)$

(b) $w_{|\mathbf{w}|} = (|\mathbf{x}|, |\mathbf{y}|)$

(c) The warping path is continuous.

(d) The warping path must be monotonic.

In particular, we are looking for a warping path that gives the minimum normalized total warping distance between $\mathbf{x}$ and $\mathbf{y}$.

If we use a Euclidean metric to define the distance between any $(\vec{x}, \vec{y})$ as

$$DIST(\vec{x}, \vec{y}) = \sqrt{\sum_{n=1}^{N} (x(n) - y(n))^2}$$

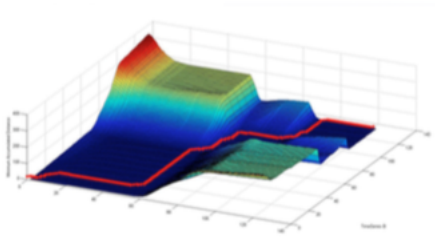
then the minimum normalized warping distance can be expressed as

$$DTW(\mathbf{x}, \mathbf{y}) = \frac{1}{|\mathbf{w}|} \sum_{k=1}^{|\mathbf{w}|} DIST(\mathbf{w}_{k_i}, \mathbf{w}_{k_j}))$$

The mininum total warping path can be found by using dynamic programing to fill a 2-D cost matrix, C, where each cell in the matrix represents the minimum warping cost to reach that point from the start. Under the rules given by our constraints, the cell value can be expressed by

$$C_{(i,j)} = DIST(i, j) + min(C_{(i-1,j)}, C_{(i,j-1)}, C_{(i-1,j-1)})$$

Once complete, we simply get the path by navigating the cost matrix in reverse order, as seen below:



To train a template, we record $M_g$ training examples for each of the G gestures. The gth template can be found by searching for the training example that provides the minimum normalized total warping distance when matched agains the other $M_g - 1$ training examples in that class. That is,

$$\phi_g = argmin \quad \frac{1}{M_g - 1} \sum_{j=1}^{M_g} DTW(\mathbf{X}_i, \mathbf{X}_j), \qquad 1 \leq i \leq M_g, i \neq j$$

To find the best therblig, we will simply get the the lowest cost from a comparison of the input with multiple time series, or output none if it doesn't reach a certain threshold.

$$min \quad DTW(\mathbf{x}, \mathbf{g})$$

for each template $\mathbf{g}$ in our set of gestures.

Nick Gillian's C++ Gesture Recognition Library has an implemetation of DTW.

3. Hand Activity Level

   The Akkas paper offered a method of quantizing the Hand Activity Level by fitting a linear regression equation as a function of root mean square (RMS) hand speed and duty cycle. In their work, hand speed is measured via examination comparison of pixel windows featuring the region of interest (hand). HAL expressed linearly was:

$$HAL_{lin} = b_0 + b_1 * S + b_2 D$$

   To scale from 0 to 10, they used the following sigmoidal logit-linear regression:

$$ln \quad [\frac{\frac{HAL}{10}}{1 - \frac{HAL}{10}}] = b_0 + b_1 ln(S) + b_2 D$$

   solving, we get the following expression for HAL

$$HAL_{log} = 10[\frac{e^{b_0 + b_1 ln(S) + b_2 D}}{1 + e^{b_0 + b_1 ln(S) + b_2 D}}]$$

For the coefficients, $b_0 = -15.87, b_1 = 2.25, b_2 = 0.02$, they achieved a $R^2$ of 0.97. Below are the predicted HAL levels from the regression:

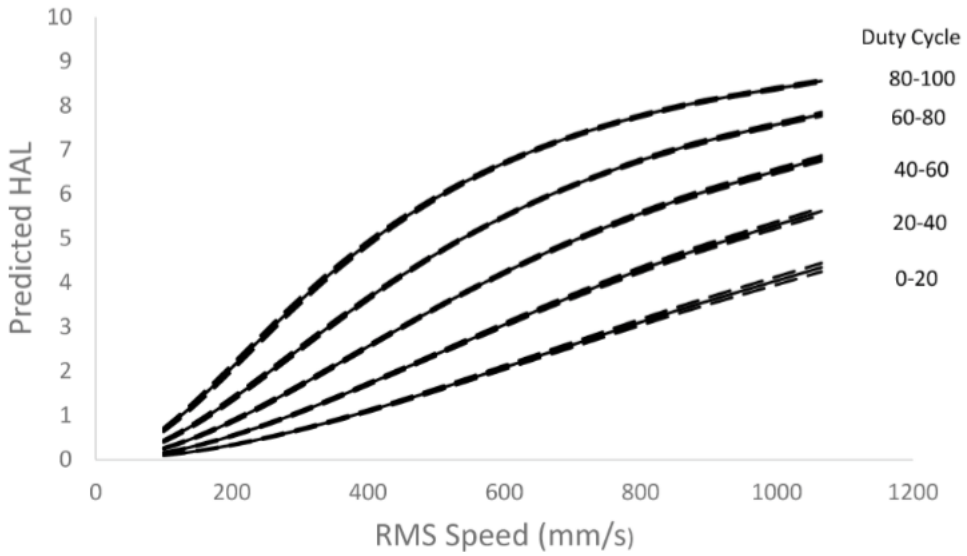



**Figure 5.**
Plot of the logit-linear equation for predicted HAL as a function of speed (S) and duty cycle (D). The solid line is the average regression line and the dotted lines represent the 0.1 percentile and 99.9 percentile of predicted HALs resulting from the Monte Carol process.

# 5 Planning - Therblig Recognition

## a) Calibration

The current implementation is to apply the Gesture Recognition Toolbox implementation of DTW to the raw EMG sensor output. This is not extendable for the following reasons:

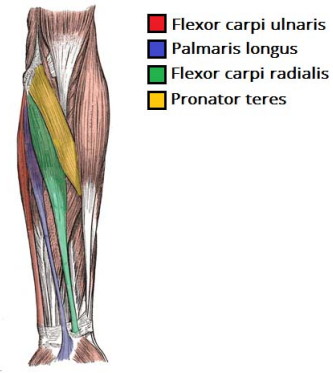
Below is a label image of the EMG sensors.

We can see that the indexing is fixed relative to itself. Therefore the armband would have to go on the arm in the exact, discrete placement each time, and also have to read equivalently across all people's arms. DTW, which only handles temporal shifts in the data, would not be able to handle this type of variance.

An ideal scenario would be that the individual user would be able to calibrate their input per trial, since we can expect the armband placement to be highly variable between trials, and the EMG readout to be distinct across different people. Calibration would allow us to standardize the input for comparison to the training set configured for the DTW algorithm.

I tentatively propose two calibration schemes, one for orientation and one for EMG normalization.

1. Orientation - It is possible we can calibrate this by having the user open and close their hand repeatedly. We could expect to see a gradient of EMG values from the *Flexor carpi ulnaris* to the *Pronator teres*. This would give us the relative position of each EMG sensor in the unit circle around the arm.




   Unfortunately, the EMG values are discrete and the armband expandable, and so there might be a slight shift on the exact position of the EMG sensor from trial to trial. If we can identify a maxima on the forearm, perhaps we can use interpolation to give us an interpolated value of the EMG at set coordinates and the exact location of the actual EMG sensors.

2. Normalization - if we can observe the interpolated maximum EMG value during the orientation phase, we could divide all values by this for a normalized input into the DTW algorithm.

## b) Learning

The following Therbligs are listed as Physical:

| Therblig | Parameters |
|---|---|
| *Physical* | |
| Transport Empty | pos. $(x,y,z)$, orient. $(\varphi,\theta,\psi)$, ang. $(\varphi',\theta',\psi')$, arm (*int*) |
| *Reaching for an object with an empty hand (precursor for Grasp)* | |
| Grasp | grasping position, grasp effort, arm |
| *Grasping an object with the active hand* | |
| Transport Loaded | goal position, orientation, angle, arm |
| *Moving an object (grasped) using a hand motion* | |
| Release Load | arm |
| *Releasing control of an object (letting go)* | |
| Hold | duration, grasp effort |
| *Hold a grasp (while other hand performs a use or an assemble function)* | |
| Position | position/orientation, arm |
| *Positioning and/or orienting an object in location (may be part of Transport Loaded)* | |
| Preposition | position/orientation, arm |
| *Positioning and/or orienting an object for the next operation* | |
| Rest | position/orientation, duration |
| *Resting to overcome fatigue, consisting of a pause in the motions* | |

The current code is training for the therbligs: Transport Empty, Grasp, and Transport Load.

Arguably, only Grasp and Release Load would have distinct DTW values.

Hold and Transport Loaded would share the same DTW value, distinguished from each other by the absense or presence of accelerometer values.

The same applies for Transport Empty and Rest. We would exect only to see a change in acceleration value, not EMG value.

The following questions remain:

1. Which therbligs are vital to identify, and which ones should be identified using DTW?
2. How do we identify a real-time time series window to feed into the DTW algorithm?

# 6 Planning - Hand Activity Level

From the current code, it seems that the current method is calculating the moving average of the acceleration and using it to update the velocity. This value is then fed into the equations with the given values from the regression fitting given above..

A moving average was probably used to accomodate due to raw acceleration output from the sensor.

Further work(?):

This might be working - I have not yet had the chance to setup and test it. How does this fit into the bigger picture?