

Principal Component Analysis used for evaluating separability of features extracted from raw EMG signal.

Description of problem

Electromyography (EMG) is widely used as control input for functional prosthetics. Using classification as control scheme has shown a low error rate when testing hand movements in a clinical environment in a single limb position. However, a challenge occurs when changing limb position as the EMG alters when performing the same hand movements in a different limb position. A newer control scheme used is linear regression, which has shown robust proportional and simultaneous control in a single limb position. We will investigate if a linear regression-based control scheme can yield similar performance across limb positions. The raw EMG signal itself is not used as control input, but features extracted from it. In this study the commonly used Mean Absolute Value (MAV) feature, and the Logarithmic Variance (LogVar) feature, which has shown linear properties in studies, will be tested as control input for four different movements (extension, flexion, ulnar deviation and radial deviation). Before the linear regression models are built, it needs to be evaluated if the features extracted from the EMG signal (eight channel EMG) for each movement are separated enough for the models to distinguish between each movement.

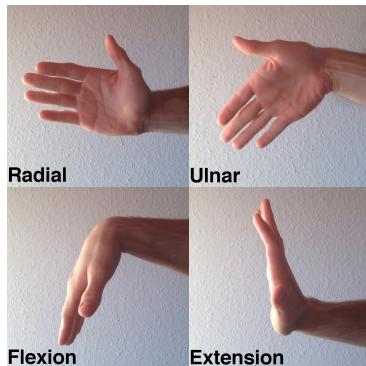


Figure 1: Hand gestures used when recording EMG signals.

Method of choice

To solve this problem we choose to implement Principal Component Analysis (PCA). PCA is an analysis tool, used to find the most defining variables in a dataset. In this case by reducing an eight dimensional feature space to a three dimensional. Thus, being able to visualize the data and quantitatively evaluate the separability of the movements.

PCA is used to express a set of possible correlated variables into uncorrelated components, called principal components. A dataset of many variables can thus be expressed in a reduced dimensionality hyperspace using less variables that are the most defining for the given dataset. Each principal component (PC) is orthogonal on the former and are uncorrelated and have zero covariance. They each define the largest variance in an axis, such that PC 1 describes the direction of the maximum variance of the dataset. Each following PC describes the next highest variance of the dataset, with the constraint that is orthogonal

and has zero covariance with any of the former PCs. PCA is the orthogonal projection of data onto a lower dimension linear space. A PC is found by minimizing the variance by projecting the feature values (red dots) onto the line describing the highest variance in the data set (purple line) as seen on figure 2. The PC (purple line) is found by minimizing the mean square distance between the data points.

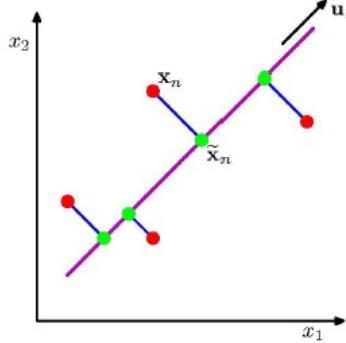


Figure 2: Projection of data variables (red dots) onto principal component line (purple)

The algebraic method of calculating the PCs is by using Singular Value Decomposition (SVD). The first step is to compute the squared cross product matrix of variances and covariances among every pair of the variables in the data set, where the diagonals are the variances and the off-diagonals are the covariances, as done in the following equation:

$$S = X'X \quad (1)$$

Where S is the cross product and X is the dataset matrix. When finding the PCs it includes an eigen-analysis of S. The eigenvalues of are solutions to the following equation:

$$|S - \lambda| = 0 \quad (2)$$

Where λ is the variances of each PC and I is the identity matrix. After solving for λ the eigenvectors can be solved through the following equation:

$$\det|S - \lambda|b_i = 0 \quad (3)$$

Where b_i is used to calculate the eigenvectors as in:

$$u_i = \frac{b_i}{\sqrt{b_i'b_i}} \quad (4)$$

Where u_i is the i number of eigenvectors that contain a contribution to the principal components. The SVD orders the eigenvalues by size $\lambda_1 > \lambda_2 \dots > \lambda_M$. The scores for each PC is equal to the corresponding eigenvalue for that exact axis. The eigenvalues describe how much of the variance is accounted for by the associated PC. Summation of all eigenvalues accounts for the total variance of the data set; this is called the trace. To find how much the each PC accounts for, the eigenvalue of that PC is divided by the total variance: $\% \text{ of total variance} = \frac{PC}{Trace}$. This can be used for deciding how many components are significant and by how much the dataset can be reduced. In our case this choice is made, due to the intention of visualizing the PC's in a three dimensional space. MatLab code for implementation is shown in the appendix after the results.

Results

When applying PCA on the MAV features for the four movements it is seen that three components describe 93.8 % of the data set. Plotting the three first PCs for each movement illustrates a separability between the movements, see figure 3. Thus, the MAV feature extracted from the acquired data can be used to build linear regression models that distinguish between the four movements. The same separability is found for the LogVar feature, see figure 4, where the first three PCs accounts for 93.97 % of the total variance.

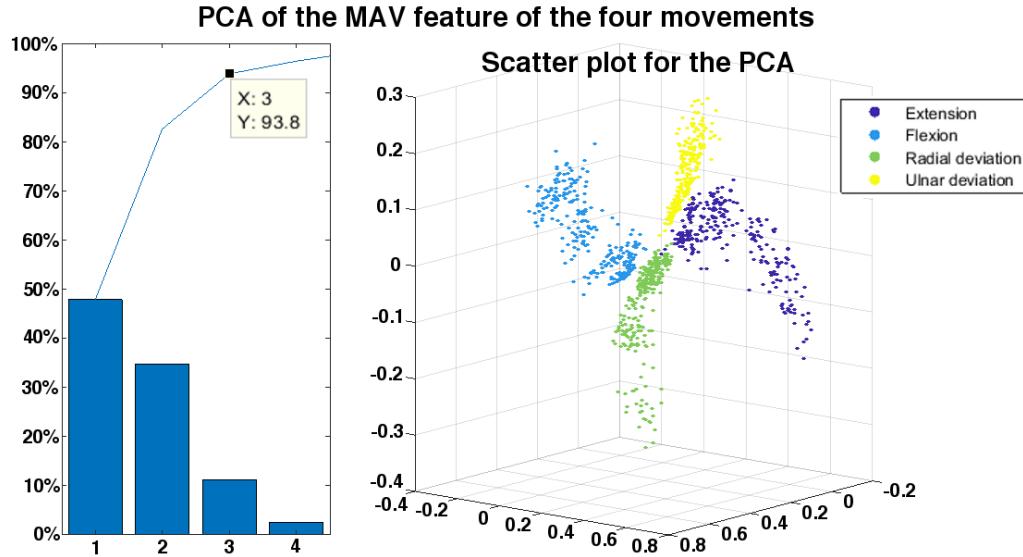


Figure 3: PCA of MAV features

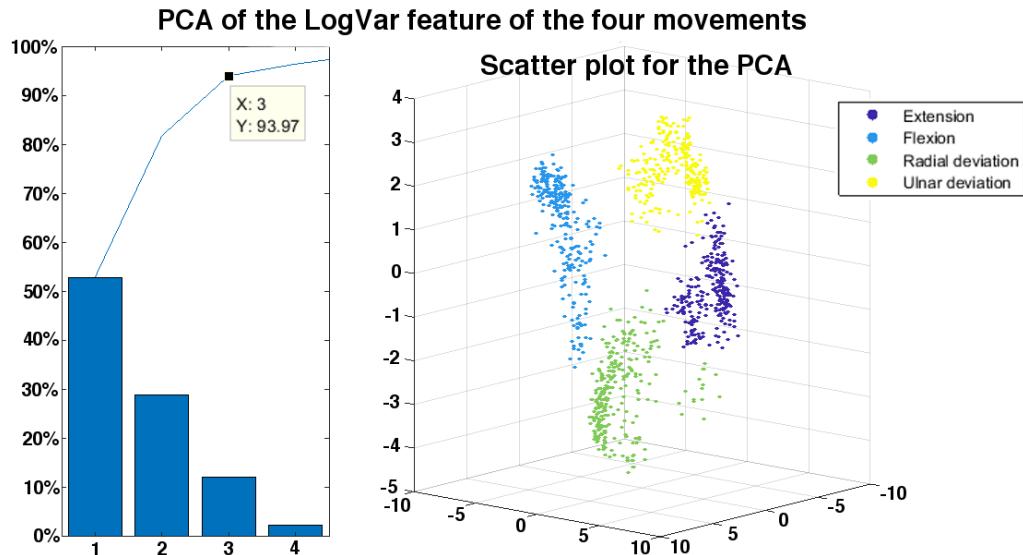


Figure 4: PCA of LogVar features

Appendix: MatLab implementation

```
function goScatterPlot(bigMatrix)

L = length(bigMatrix)/4;

Ex = ones(L,1)*0;
Fl = ones(L,1)*1;
Ra = ones(L,1)*2;
Ul = ones(L,1)*3;

overallMovement = [Ex;Fl;Ra;Ul];

[pcs, scrs, ~, ~, pexp] = pca(bigMatrix);

figure(1)
%Intensity of principal components
subplot(1,3,1)
pareto(pexp);
xlabel('a)')
title('PCA of the LogVar feature of the four movements')
subplot(1,3,[2 3])
%Scatterplot of the four movements
scatter3(scrs(1:L, 1 ), scrs(1:L, 2 ), scrs(1:L, 3 ), 10, ...
    overallMovement(1:L,1), 'filled')
hold on
scatter3(scrs(L+1:L*2, 1 ), scrs(L+1:L*2, 2 ), scrs(L+1:L*2, 3 ), ...
    10, overallMovement(L+1:L*2,1), 'filled')
scatter3(scrs(L*2+1:L*3, 1 ), scrs(L*2+1:L*3, 2 ), scrs(L*2+1:L*3, 3 ), ...
    10, overallMovement(L*2+1:L*3,1), 'filled')
scatter3(scrs(L*3+1:L*4, 1 ), scrs(L*3+1:L*4, 2 ), scrs(L*3+1:L*4, 3 ), ...
    10, overallMovement(L*3+1:L*4,1), 'filled')
legend('Extension', 'Flexion', 'Radial deviation', 'Ulnar deviation')
xlabel('a)')
title('Scatter plot of the PCA')
```