GAIT PATTERN ANALYSIS USING EMG via MACHINE LEARNING FOR OPTIMIZING LOWER LIMB EXOSKELETON USAGE

SUBMITTED AS PART OF FINAL PROJECT
ME 6250 : WEARABLE ROBOTICS
NORTHEASTERN UNIVERSITY BOSTON

VARUN RAGHAVENDRA
MS ROBOTICS

INDEX

1. **Introduction and Motivation**
2. **Methodology**
3. **Results and Discussion**
4. **Conclusion and Future Scope of Work**

Introduction and Motivation

All the code and data is available in the following Github Repository :
https://github.com/varuncraghavendra/Wearable_Robotics.git

Abstract : The aim of this project is to make machine learning models understand the distinct and underlying features of sEMG signals from lower limb muscles, in time and frequency domain, further use binary classification models to assess if the gait is normal or abnormal, to estimate the need for exoskeleton assistance.

Introduction to Exoskeleton Devices : Assistive wearable devices used for enhanced mobility, in clinical rehabilitation and musculoskeletal training.

Introduction to EMG : Physiological signals signifying muscle activation of the muscles.

MOTIVATION : Understanding muscle synergy and narrowing down recovery time for lower limb rehabilitation using exoskeletons, by targeted assistance to muscles associated with knee mechanics.

Previous studies have shown that integrating EMG into exoskeleton-based rehabilitation improves recovery efficiency, particularly in conditions like:

- Post-surgical rehabilitation (e.g., ACL repair): Targeting knee extensors like RF and VM.
- Neurological impairments (e.g., stroke, spinal cord injury): Restoring coordination between knee flexors and extensors (BF/ST vs. RF/VM).
- Age-related muscle decline: Enhancing muscle reactivation to prevent atrophy.

Hypothesis made in the project :

Learning patterns of muscle activation (via **EMG**) using **ML** and providing closed-loop **exoskeleton** assistance could be beneficial in :

- Encouraging volitional control
- Fall prevention
- Track progress of recovery post therapy during rehab.
- Providing flexibility for diverse conditions

Methodology

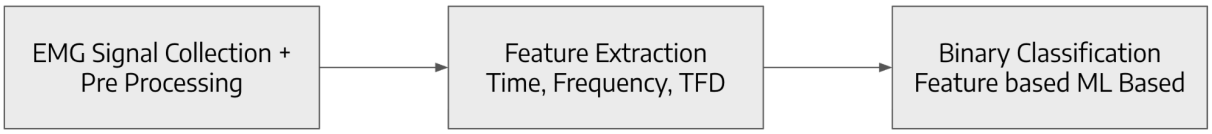

Figure 1. Flowchart of the project (Each muscle is subjected to each block)

Signal Collection :

A typical approach to collect surface EMG data from multiple lower limb muscles at the same would be as follows :

- Choose a surface EMG data collection device that provides multiple analog and digital channels (for eg. MWX8 by Biometrics)
- Choose an appropriate sampling frequency for the recording (typically it is 1000-2000 Hz)
- Skin Preparation : Skin Cleaning, Hair Removal, Scrubbing of skin, Applying gel
- Electrodes : Ag/AgCl Electrodes for sEMG signals.

In this project, I have used an open source dataset published on UCI Machine Learning Library

Reference : https://archive.ics.uci.edu/dataset/278/emg+dataset+in+lower+limb

Highlights of the dataset :

Test Subjects : 22 male subjects, 11 with different knee abnormalities previously diagnosed by a professional.

Device Used : MWX8 by Biometrics
Type of EMG : Surface (Raw)
No. of Channels : 4 [Biceps Femoris, Rectus Femoris, Semitendonous, Vastus Medialis]
Sampling Frequency : 1kHz Additional Data : Knee Goniometer (1) (Not used in project)

Signal Pre Processing

Signal Normalization :

I have used Maximum Voluntary Contraction (MVC) due to its advantage of comparability with other test subjects or conditions.

Other types of normalization of sEMG signals

Min-Max Normalization : Sensitive to outliers
Z-Score Normalization : The data can go out of range
RMS Normalization : Can smooth out rapid signal changes

Signal Filtering :

sEMG signals from lower limb muscles usually lie between 10-500Hz, thereby we need to remove noises and artifacts from the muscles.

For this project, I have designed a 4th Order Bandpass filter, with a range of 10/20 Hz to 450 Hz.

I have also introduced a notch filter. It is important too (@50Hz/60Hz) to remove the electrical noise coming from the power lines (@plugpoints / power sources)

Signal Rectification :

Converting all negative values of EMG to positive, to create an envelope to easily interpret the signal, and segment the signal better based on time. It is a great indicator of the amplitude of the EMG signal too.

Signal Smoothing :

Reduces high frequency noise and fluctuations in EMG signals.
In this project, I am using a moving average technique for smoothing, with a window size of 50ms, using the movmean function of MATLAB.

Feature Extraction

I used this paper to select and code up the features [Ref]
The features were collected from each of these muscles
[Biceps Femoris, Rectus Femoris, Vastus Medialis, Semitendinosus]

Time Domain : 11 Features

- Mean Value
- Standard Deviation
- Integrated Absolute Value
- Mean Absolute Value
- Enhanced Mean Absolute Value
- Waveform Length
- Average Absolute Value of change
- Root Mean Square Value
- Modified Mean Absolute Value
- Logarithmic Detector
- Slope Sign Change

Frequency Domain : 6 Features

- Mean Frequency
- Median Frequency
- Peak Frequency
- Total Power
- Spectral Entropy
- Peak to RMS Ratio

Time-Frequency Domain (Discrete Wavelet Transform) : 3 Features

- Wavelet Energy
- Wavelet Entropy
- Maximum Wavelet Coefficient

Binary Classification

As discussed in the aim, we have two classes of data (normal gait and abnormal gait)
I have labelled the normal gait as '0' and abnormal gait class as '1' in my code.

I chose the following algorithms to validate the binary classification accuracies :

K-Nearest Neighbour (KNN) :

- KNN is a model that performs well when trained on low-dimensional and well separated data, and the performance could degrade for high-dimensional data.
- There is no prior assumptions made on the data distribution (non-parametric)
- It could be sensitive to outliers and computationally expensive for larger datasets

Decision Tree (DT) :

- Easy to visualize
- Performs well on continuous and categorical features
- Handles non-linear data well
- Prone to overfitting

Random Forest (RF) :

- Robust to Overfitting
- Works well with high-dimensional data
- Takes more time and computation than single decision tree

SVM :

- Can handle complex and high-dimensional data
- Works well with small and medium sized datasets
- Handles non-linear data well
- Sensitive to parameter selection (kernel and regularization)

NOTE : I also learnt that the features of EMG time and frequency domain features have high linearity, although I tested the model for non-linear parameters and kernels too, which is explained in the further section.

Optimization Techniques

Dimensionality Reduction :

Since there were many parameters chosen for training, the model was visibly overfitting with absurd 100% accuracy across some models. Hence, I decided to perform a dimensionality reduction to control overfitting and reduce computational cost.

I chose linear PCA (Principal Component Analysis) for dimensionality reduction, due to the nature of linearity in both time and frequency domain features of EMG signals.

The further details that I used about PCA are as follows :

- Variance Threshold : I tried 99% and 95%, and 95% showed consistent high accuracies across trials
- Standardization of Features : I applied a z score normalization on features, so that they have zero mean and unit variance.
- Then, I estimated a cumulative variance of features as the principal components were added. Then, I started comparing the variance with the threshold chosen earlier to retain the smallest number of components required to retain the variance.
- Finally, construct the reduced feature set using the principal components.

K-Fold Cross Validation :

Cross validation is a necessary step to prevent overfitting and handle variability in data.

Data has to be prepared first (split into training and testing data) and then I applied the MATLAB function (cvpartition) to stratify the data. Furthermore, the predictions were made after a ‘4’ fold classification, ‘5’ fold and ‘10’ fold cross validation, although it seems that I obtained the best and consistent results from ‘4 Fold Cross Validation’

Precision, Recall and F-1 Scores :
These parameters were calculated to gain insights on how well the model is performing across different aspects of the classification.

Train : Test Ratio = 70% Training Data; 30% Test Data

Hyperparameter Tuning :

KNN : Single parameter, k (number of neighbours). I tried out 5,7,9,11 and observed that the value '5' provided the best and consistent values across trials.

Decision Tree (DT) : Single parameter : Decision Tree Max Splits
I tried 10,20 and 50 and got the best and consistent values across trials at the value '20'

Random Forest (RF) : Two parameters :

No. of Trees in the Forest : I chose this parameter to be '100' is a common starting point for providing robust predictions.
Minimum Leaf Size : I chose the leaf size to be 1, which allows the model to capture finer details in the data, which could be beneficial in small datasets too! Larger values of this parameter could smoothen out predictions.

Support Vector Machine (SVM) : Two Kernels tested (Linear and Radial Basis Function) One is a linear kernel and one is a non-linear kernel, I used both to examine the linearity of the data.

Further parameters of SVM included :

Box Constraint : The regularization parameter controlling the trade-off between maximizing margin and minimizing classification error. I chose it to be '1' which gives it a balanced tradeoff.

Gamma : This parameter determines the influence of a single training example in non-linear kernels. A moderate gamma value ensures a balance between bias and variance. Lower values lead to smoother decision boundaries. When I used Radial Basis Function (rbf) as the kernel function, gamma was used. I tried out gamma values of 1, which gave me great and consistent results. Smaller values of gamma could lead to smoother decision boundaries, which could turn out to be good for EMG data with large scale trends. Although, I focused on optimizing the model using the linear kernel.

Results and Discussions

Best Results - Project

Classifier	Feature Type	Mean Accuracy
KNN	TD	62.5
KNN	FD	75
KNN	Wavelet	62.5
SVM	TD	93.75
SVM	FD	68.75
SVM	Wavelet	75
Random Forest	TD	81.25
Random Forest	FD	56.25
Random Forest	Wavelet	62.5
Decision Tree	TD	62.5
Decision Tree	FD	43.75
Decision Tree	Wavelet	75

Biceps Femoris

Classifier	Feature Type	Mean Accuracy
KNN	TD	68.75
KNN	FD	56.25
KNN	Wavelet	56.25
SVM	TD	62.5
SVM	FD	56.25
SVM	Wavelet	56.25
Random Forest	TD	56.25
Random Forest	FD	68.75
Random Forest	Wavelet	62.5
Decision Tree	TD	43.75
Decision Tree	FD	62.5
Decision Tree	Wavelet	50

Semitendinosus

Classifier	Feature Type	Mean Accuracy
KNN	TD	68.75
KNN	FD	62.5
KNN	Wavelet	56.25
SVM	TD	62.5
SVM	FD	68.75
SVM	Wavelet	56.25
Random Forest	TD	81.25
Random Forest	FD	75
Random Forest	Wavelet	50
Decision Tree	TD	68.75
Decision Tree	FD	87.5
Decision Tree	Wavelet	56.25

Rectus Femoris

Classifier	Feature Type	Mean Accuracy
KNN	TD	75
KNN	FD	56.25
KNN	Wavelet	68.75
SVM	TD	75
SVM	FD	75
SVM	Wavelet	81.25
Random Forest	TD	75
Random Forest	FD	56.25
Random Forest	Wavelet	68.75
Decision Tree	TD	75
Decision Tree	FD	62.5
Decision Tree	Wavelet	75

Vastus Medialis

Discussion about classification results :

From the above classification results, it can be clearly observed that :

Overall Feature Type Performance:

- Time Domain features generally outperform Frequency Domain and Wavelet features.
- Random Forest and SVM are more consistent across metrics.

Overall Classifier Performance:

SVM and Random Forest tend to achieve the highest scores, especially when paired with Time Domain features.

Hypothesis from the results : It is a positive sign that time domain features are distinguishable, since the exoskeleton devices can be manipulated in real time using time domain features. It is also true that the extraction of frequency domain features take time for processing (fourier transforms) and hence it can not be used in closed-loop real time exoskeleton.

Conclusion

From the above observed results, I can conclude that for sEMG data for gait classification of lower-limb (quad and hamstring) muscles, Random Forest and SVM tend to provide the best performance in terms of accuracy and robustness across varied conditions. Decision Trees and KNN can still be useful, especially for simpler or smaller datasets, but they may not handle the complexity and noise in sEMG data as effectively as ensemble methods or SVM.

Future Scope of Work

- Explore light weight deep learning models for real time adaptive systems
- Gait Phase Estimation by learning movement patterns, with additional sensors (IMU/Goniometer/Pressure Sensors)
- Explore time-frequency representations to capture subtle variations in gait patterns through wavelet analysis and entropy measures
- Extension of research to multiple muscles and age category based validation.