**Investigation of MATLAB Neural Network Patterning for EMG Classification**

**Nicolas Fernandez1  Jordan Hayes2**

1 Engineering Student, Grand Valley State University, Grand Rapids, United States

[fernanic@mail.gvsu.edu](mailto:fernanic@mail.gvsu.edu)

2 Engineering Student, Grand Valley State University, Grand Rapids, United States

[hayejord@mail.gvsu.edu](mailto:hayejord@mail.gvsu.edu)

Word Count: 2642

16 Figures and 1 Table(s)

Received: October 28, 2021

EGR 434-01 Bioelectric Potentials

**Abstract**

Electromyography is the study of the electrical activity in muscles. EMG data can be used in a wide variety of ways to help rehabilitate or diagnose muscle disorders. The objective of this study is to train a backpropagation neural network to classify subjects from their EMG data as healthy, an ALS patient, or another form of myopathy by means of classification using a neural network. MATLAB processing and deep learning Neural Pattern Recognition Tool was used to present data, train, and analyze performance. Results indicated best performance (~84% correct in testing) using autoregression coefficients as input, 20 neurons, 70-15-15% data allocation to training phases, and using the Cross Entropy performance function. With all these factors accounted for in the optimal network, the network performed to a statistically significant level, and the network outperformed a 99% significance level by several orders with a *p* value less than 0.00001.

Introduction

Electromyography is the study of the electrical activity in muscles. EMG data has been used in analysis of rehabilitation, gait analysis, motor control, ergonomics, fatigue analysis, and diagnosing muscular diseases. Amyotrophic lateral sclerosis (ALS) is a progressive and inevitably fatal example of these muscular diseases. ALS is the most common adult motor neurone disease (MND) worldwide and is considered the prototypical disorder of this class [5]. Diagnostic accuracy of muscle

biopsy and EMG in patients with myopathy varies [4]; therefore, there has been a concerted effort to increase the accuracy of diagnosis techniques between ALS and other myopic disorders. This study will focus on training a neural network to differentiate and classify ALS patients from healthy control subjects and from a separate group of myopathy patients. Data was recorded with concentric needle electrodes, boosted with gain and filtered with a bandpass filter, sampled by a 16-bit ADC, and is stored as binary files to be loaded into MATLAB.

Methods

The data collected for this experiment was taken from EMGLAB’s database from Dr. Miki Nikolic’s dissertation on EMG signal decomposition [2]. There were three groups of subjects that had EMG data taken. A healthy control group, a myopathy group, and an ALS group with 10, 7, and 8 subjects, respectively.

Selection and Description of Participants

The participants involved in the study had many different muscles measured for EMG data, but the Biceps Brachii was taken on the most subjects. Only two subjects were omitted from the dataset because the database had no EMG data for their biceps. Table 1 contains a basic description of each subject and their diagnosis from the study.

Table 1. Subject Descriptions and Diagnosis (A2 and M1 Omitted)



From the table, it is important to note that there may be a confounding variable of age as the healthy subjects have a large age gap from the ALS subjects.

Data Collection

The EMG data was collected from a macro needle electrode that combined with a concentric needle [2]. A macro needle is a mentioned to be a poor indicator of total activity in a motor unit, but recording with more length gives more surface area and improves the sampling method by encompassing a larger area of 15 mm. When combined with a concentric needle, better amplitude motor unit action potentials were recorded across age groups and conditions [2]. The ideal firing pattern to measure when using conventional needles like concentric needles are low contractions, and that signal is measured in a constant isometric contraction for the original purpose of obtaining mean firing rate and variability [2]. Figure 1 gives a helpful diagram of the EMG signal collection and manipulation before it was fed into MATLAB.

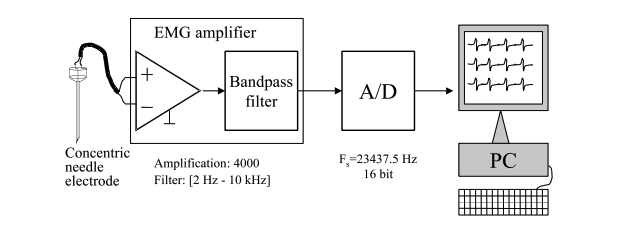


Figure 1. Signal Acquisition Overview [2]

In Figure 1, the concentric needle is shown feeding into an amplifier circuit that gives the signal a 500µV/division gain and then passes through a 2 Hz – 10 kHz bandpass filter. The analog signal is then processed by a 16-bit ADC sampling at 23,437.5 Hz to convert the digital signal that was stored in the database. The signal amplitudes varied such that normalization was necessary because otherwise the signal variation ratio between subjects was as large as 200. The foundation of the neural network was constructed from the input of these signals.

Defining a Neural Network

Artificial Neural Networks are made up of an input layer, one or more hidden layers, and an output layer. The input layer for this study is a single row vector from which to train the hidden layers. The hidden layers all have associated weights and biases, similar to a linear regression model. Adjusting the weights of each neuron is the aim of training algorithms. Mean square error (MSE) is the most prominent criterion in training neural networks and has been employed in numerous learning problems. MSE measure is based on the following equation. This application of MSE is different from the performance function.

(1)

The neural network was trained in MATLAB, which uses a scaled conjugate gradient (CG) backpropagation (BP) algorithm modelled after a new BP method that is more robust than the MSE-based method in the sense of impulsive noise, especially when SNR is low.

A picture containing text, clock

Description automatically generated

**Figure 2. MATLAB Neural Network Block Diagram**

In general, BP methods adjust the weights of neural networks to minimize MSE between the actual output samples and the desired data set by shifting the error backwards layer by layer and attributing the weights according to error. The sequence of weights is generated using the equation below

(2)

where α is the learn ingrate or step size, is the weights vector of the previous iteration, and is the gradient vector and is calculated by

(3)

Where is the MSE error of the outputs in the kth step of the training procedure.

Strategies for Training the Network

The training of the network has 3 phases: training, validation, and testing. Training phase is used to give the network information to make decisions; validation ensures the training is sufficient and ensures the training data was not skewed or poorly representative; the testing phase tests the network’s ability to correctly classify the signals. Any percentage of the entire dataset can be assigned to each phase as long as it totals 100%. The network also has a performance function used to evaluate performance over time. There are many feature extraction techniques that were used to extract the signals such as Autoregressive Coefficients (AR), Root Mean Square (RMS), and Mean Absolute Value (MAV) [1]. These signals were presented to the network for training. Finding the best performance capability of the network was then an iterative process, continually changing the signal characteristic that was presented, performance function of the network, percentage of data used for each phase of training, and number of neurons in combination. It is necessary for some important neural network characteristics to be explained.

Cross Entropy is defined in signal processing as the difference between two probability distributions for a given random variable, and in this case, are the target distribution and network’s approximation of the target distribution. The important characteristic about entropy is that it will increase when the system encounters unpredictable information and is therefore able to learn more from the information. Entropy will decrease when events become more predictable to the system, and the difference between the actual target and the approximation distributions is low.

Overfitting occurs when the statistical model perfectly fits the training data. A common source of this problem is too many neurons being used to predict the data. When this happens, cross entropy will be very high when new data is presented, simply because it is not identical to the training data. The algorithm will not perform well against unseen or “surprising” data. If the cross entropy decreases below a threshold for a certain number of iterations, the Network Patterning Tool will stop running as a caution of overfitting when it has a good measure of predictability.

Standard error is the numerical difference between the target and the network’s approximation. The lower the error (but non-zero), the more accurate the approximations are, and the network is more likely to classify a dataset correctly.

Statistical Significance of Results

The statistical significance of the neural network output was calculated in order to evaluate the network’s performance in relation to if it were simply guessing. The EMG classifications can be treated like a Bernoulli trial wherein the network either identifies the signal to the correct group or not. This binomial distribution has a probability of guessing a random EMG signal correctly to be *p* = 0.33 because there are three groups and inversely *q* = 0.66 as a probability of guessing incorrectly. The following equations establish the expected mean and standard deviation when testing 324 samples.

(4)

(5)

The expected value for the network operating by random chance is 108 correct classifications with standard deviation of 8.485 correct classifications. With these Bernoulli trial values, a one-tailed Z Test can be conducted to find the chances of performing better than the minimum correct classifications at any stage of the neural network of 70% (226.8 correct classifications). The equation for a Z score is shown below.

(6)

Where the is the correct classifications of the network of 226.8, µ is the expected correct classifications of 108, is the variance of the dataset of 72, and is the dataset size of 324. A one-tailed Z test was performed after assessing the network’s output.

Results

All results are documented in relation to the best-performing network. The settings of the optimal network were a cross entropy performance function, 20 hidden neurons, EMG autoregression coefficients for input, and a data split of 70% of the samples for training, 15% of the samples for validation, and 15% of samples for testing. The confusion plot describes the percent of data that was classified correctly and incorrectly at each stage of the creation of the neural network.

A picture containing text, cabinet, lined, different

Description automatically generated

Figure 3. Confusion Plot of Optimal Network

The figure above shows the confusion matrix of the optimally performing data presentation and number of network neurons. The testing phase resulted in 71%; the validation phase, 78%; the testing phase, 84%.

Chart, line chart

Description automatically generated

**Figure 4. ROC Plot for Optimal Network**

Figure 4 shows the ROC plot for the optimal network. All results are bowed highly toward the top left corner. The top left corner is a true positive rate of 1 and represents a network that perfectly describes the data. In the testing phase, the best performing class was the myopathy group.

Chart

Description automatically generated

**Figure 5. Error Histogram of Optimal Network**

The figure above shows the error histogram of the optimal network. The highest errors occur in testing and the errors are normally distributed, skewed left. The largest bin represents the number of classifications with very little to zero error.

Now results will be presented for other networks where one of the variables were changed from the optimal network to show the effects of each variable.

A picture containing text, cabinet, lined

Description automatically generated

**Figure 6. Confusion Plot with Changed Data Allocation to 30-35-35% Respective to each Training Phase**

Figure 6 shows the confusion matrix of a network trained with 30% of the data, validated with 35%, and tested with 35%. The results of the training, validation, and testing phases are 49%, 51%, and 47%, respectively.

Chart, bar chart

Description automatically generated

**Figure 7. Error Histogram of Data Allocation to 30-35-35% Respective to each Training Phase**

The error histogram above shows the error calculations from the network. It can be seen that the plot is not nearly normally distributed, and an equal number of instances are recorded for many of the bins in the distribution.

A picture containing text, cabinet, lined, different

Description automatically generated

**Figure 8. Confusion Plot of Training Network with RMS Vector as Input**

The figure above shows the confusion matrix of a network trained with RMS values instead of Autoregression coefficients. Results show 48% correct in training, 47% in validation, and 49% in testing.

A picture containing text, cabinet, lined

Description automatically generated

**Figure 9. Confusion Plot with MSE Used as Performance Function**

The figure above shows the confusion matrix of a network trained using MSE as the performance function. Results show 70% in training, 69% in validation, and 63% in testing.

A picture containing text, cabinet, lined

Description automatically generated

**Figure 10. Confusion Plot Using Only 5 Neurons**

Figure 10 shows the confusion matrix of a trained network with only 5 neurons. The training and validation results in 69% correct, while the testing phase results in 47% correct.

Chart

Description automatically generated

**Figure 11. ROC Plot Using Only 5 Neurons**

The figure above displays the Regional Operating Characteristic (ROC) output for training a network with only 5 neurons. The training phase is bowed well towards the top left corner, but the testing phase places Class 1 and 3 very close to the grey line, indicating little to no prediction capability.

After examining the configuration of the optimal network output, a one-tailed Z test was conducted. The calculated Z score using Equation 6 for a one-tailed test with a significance level of 0.01 was 252, which corresponds to a *p* value of less than 0.00001. This is the probability that the network got the results it did by random chance.

Conclusion

This paper applied new CG BP methods derived from information theoretic concepts to an algorithm for a neural network. The effects of continually changing the signal characteristic that was presented, performance function of the network, percentage of data used for each phase of training, and number of neurons in combination are shown in the results. The best performance function was Cross Entropy. Overfitting can be avoided by using regularization [6], which is a performance function in MATLAB, but the Cross-Entropy function consistently outperformed regularization. Performance of the network decreased when using different performance functions such as the MSE. The Autoregression coefficients trained the network to classify the signals best, where the error histogram is normally distributed yet skewed left. Skewness is not necessarily bad because validation performed well following the network training. Most errors are in the bin that includes zero and minimal errors. When the input was represented differently (RMS, time-domain, PSD, etc…), performance decreased. It was expected that the median frequency and PSD would represent the signals best because the frequency characteristics of EMG perform well in other analyses. The fact that this manipulation of EMG signals was not the highest performing was surprising. The number of neurons cannot be too high, or else the network would be at risk of overfitting, or too low, or else the data would be underfitted. Using 5 neurons, instead of 20 used in the optimal method, resulted in network underfitting, high entropy, and a testing ROC curve that runs along the grey “no prediction” line. The grey line represents an equally likely chance of being right or wrong, so the algorithm learned very little for that dataset. Changing the ratio of samples used in each phase also impacted the network’s effectiveness, because withholding the majority of the data from being used in the training phase severely cut performance in the testing phase because the algorithm has not been presented with enough information about the patterns within each class to make statistically significant decisions; large datasets are then needed because enough samples have to be left over for the validation and testing phases to be a good representation of general performance. With all these factors accounted for in the optimal network, the network performed to a statistically significant level. If the threshold for significance in a 99% confidence interval is a *p* value less than 0.01, the network outperformed that significance by several orders of magnitude at a *p* value of less than 0.00001.

Acknowledgements

This work was supported by the research of Dr. Miki Nikolic through the detail and care put into measuring EMG signals. The study also was given direction by the work of A. R. Heravi and Abed Hodtani when deducing which signal manipulations to conduct to get more distinct classifications. In addition, Dr. Shamita Rhodes, Professor of Biomedical Engineering, provided guidance as to the structure and execution of the study.

References

1. Heravi, A. R., & Abed Hodtani, G. (2018). A new correntropy-based conjugate gradient backpropagation algorithm for improving training in Neural Networks. IEEE Transactions on Neural Networks and Learning Systems, 29(12), 6252–6263
2. Nikolic M. Detailed Analysis of Clinical Electromyography Signals EMG Decomposition, Findings and Firing Pattern Analysis in Controls and Patients with Myopathy and Amytrophic Lateral Sclerosis. PhD Thesis, Faculty of Health Science, University of Copenhagen, 2001. [The data are available as dataset N2001 at http://www.emglab.net]
3. Artificial Neural Network: Understanding the Basic Concepts without Mathematics <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6428006/>
4. Constantitnedes, , Papahatzaki, Papadimas, Karandreas, O. S., Zambelis, Kokotis, & Manda (2018). Diagnostic accuracy of muscle biopsy and electromyography in 123 patients with neuromuscular disorders. *In Vivo*, *32*(6), 1647–1652. https://doi.org/10.21873/invivo.11427
5. Joyce, N. C., & Carter, G. T. (2013). Electrodiagnosis in persons with amyotrophic lateral sclerosis. *PM&R*, *5*. https://doi.org/10.1016/j.pmrj.2013.03.020
6. *trainbr*. Bayesian regularization backpropagation - MATLAB. (n.d.). Retrieved December 10, 2021, from <https://www.mathworks.com/help/deeplearning/ref/trainbr.html>.
7. Elamvazuthi, I., Duy, N. H. X., Ali, Z., Su, S. W., Khan, M. K. A. A., & Parasuraman, S. (2015). Electromyography (EMG) based classification of neuromuscular disorders using Multi-Layer Perceptron. *Procedia Computer Science*, *76*, 223–228. https://doi.org/10.1016/j.procs.2015.12.346
8. IBM Cloud Education. (n.d.). *What are neural networks?* IBM. Retrieved December 10, 2021, from https://www.ibm.com/cloud/learn/neural-networks.
9. Kostadinov, S. (2019, August 12). *Understanding backpropagation algorithm*. Medium. Retrieved December 10, 2021, from https://towardsdatascience.com/understanding-backpropagation-algorithm-7bb3aa2f95fd.

**Appendix A (MATLAB Figures)**

Table 1. Subject Descriptions and Diagnosis (A2 and M1 Omitted)

Table

Description automatically generated

A picture containing text, cabinet, lined

Description automatically generated

Figure 3. Confusion Plot of Optimal Network

**\**

Chart, line chart

Description automatically generated

**Figure 4. ROC Plot for Optimal Network**

Chart

Description automatically generated

**Figure 5. Error Histogram of Optimal Network**

A picture containing text, cabinet, lined

Description automatically generated

**Figure 6. Confusion Plot With Changed Data Allocation to 30-35-35% Respective to each Training Phase**

Chart, bar chart

Description automatically generated

**Figure 7. Error Histogram of Data Allocation to 30-35-35% Respective to each Training Phase**

**Chart, line chart

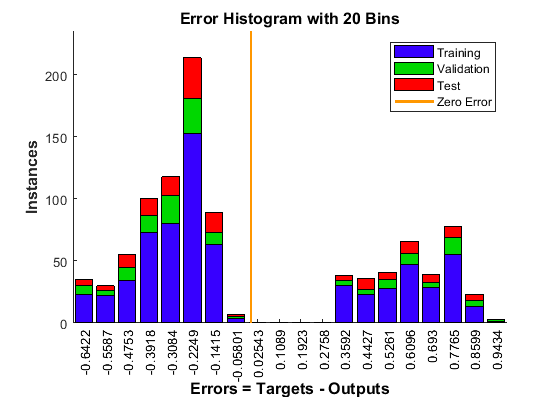
Description automatically generated**

**Figure 12. ROC Plot of Data Allocation to 30-35-35% Respective to each Training Phase**

A picture containing text, cabinet, lined, different

Description automatically generated

**Figure 8. Confusion Plot of Training Network with RMS Vector as Input**

****

**Figure 13. Error Histogram of Using RMS to Train Network**

Chart

Description automatically generated

**Figure 14. ROC Plot Using RMS to Train Network**

A picture containing text, cabinet, lined

Description automatically generated

**Figure 9. Confusion Plot with MSE Used as Performance Function**

Chart

Description automatically generated

**Figure 15. Error Histogram of Using MSE as Performance Function**

Chart, line chart

Description automatically generated

**Figure 16. ROC Plot Using MSE for Performance Function**

A picture containing text, cabinet, lined

Description automatically generated

**Figure 10. Confusion Plot Using Only 5 Neurons**

Chart

Description automatically generated

**Figure 11. ROC Plot Using Only 5 Neurons**

Chart

Description automatically generated

**Figure 16. Error Histogram of Using 5 Neurons**

**Appendix B (MATLAB Code)**

**B1. Network Setup Code (File Imports)**

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Course : EGR 434

% Instructor : Dr. Rhodes

% Author : Nicolas Fernandez and Jordan Hayes

% Assignment: ALS Detection Final Project

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% PART 1

%

% Load in EMG data for Control subjects,

% myopathy subjects, and ALS subjects.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

EMG\_Master\_TimeData = zeros(524288, 270);

EMG\_Master\_PSDData = zeros(524288, 270);

EMG\_Names = strings(1,488);

i = 1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% LOAD ALS DATA

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

EMG\_Names(i) = 'N2001A01BB01.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A01BB02.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A01BB03.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A01BB04.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A01BB05.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A01BB06.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A01BB07.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A01BB08.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A01BB09.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A01BB10.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A01BB11.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A03BB51.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A03BB52.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A03BB53.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A03BB54.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A03BB55.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A03BB56.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A03BB57.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A03BB58.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A03BB59.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A03BB60.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A03BB61.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A03BB62.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A03BB63.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A03BB64.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A03BB65.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A03BB66.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A03BB67.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A03BB68.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A03BB69.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A04BB51.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A04BB52.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A04BB53.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A04BB54.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A04BB55.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A04BB56.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A04BB57.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A04BB58.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A04BB59.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A04BB60.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A04BB61.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A04BB62.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A04BB63.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A04BB64.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A04BB65.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A04BB66.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A04BB67.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A04BB68.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A04BB69.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A05BB51.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A05BB52.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A05BB53.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A06BB01.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A06BB02.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A06BB03.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A06BB04.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A06BB05.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A06BB06.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A06BB07.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A06BB08.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A06BB09.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A06BB10.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A06BB11.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A06BB12.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A06BB13.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A06BB14.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A06BB15.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A06BB16.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A06BB17.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A07BB01.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A07BB02.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A07BB03.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A07BB04.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A07BB05.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A07BB06.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A07BB07.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A07BB08.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A07BB09.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A08BB01.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A08BB02.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A08BB03.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A08BB04.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A08BB05.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A08BB06.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A08BB07.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A08BB08.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A08BB09.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A08BB10.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A08BB11.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A08BB12.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A08BB13.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001A08BB14.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

sizeALS = i;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% LOAD CONTROL DATA

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

EMG\_Names(i) = 'N2001C01BB51.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C01BB52.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C01BB53.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C01BB54.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C01BB55.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C01BB56.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C01BB57.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C01BB58.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C01BB59.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C01BB60.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C01BB61.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C01BB62.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C01BB63.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C01BB64.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C01BB65.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

% EMG\_Names(i) = 'N2001C01BB66.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C01BB67.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C01BB68.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C01BB69.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C01BB70.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C01BB71.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C01BB72.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C01BB73.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C01BB74.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C01BB75.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C01BB76.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C01BB77.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C01BB78.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C01BB79.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

EMG\_Names(i) = 'N2001C02BB51.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C02BB52.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C02BB53.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C02BB54.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C02BB55.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C02BB56.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C02BB57.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C02BB58.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C02BB59.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C02BB60.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C02BB61.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C02BB62.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C02BB63.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C02BB64.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C02BB65.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

% EMG\_Names(i) = 'N2001C02BB66.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C02BB67.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C02BB68.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C02BB69.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C02BB70.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C02BB71.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C02BB72.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C02BB73.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C02BB74.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C02BB75.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C02BB76.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C02BB77.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C02BB78.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C02BB79.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

EMG\_Names(i) = 'N2001C03BB51.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C03BB52.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C03BB53.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C03BB54.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C03BB55.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C03BB56.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C03BB57.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C03BB58.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C03BB59.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C03BB60.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C03BB61.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C03BB62.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C03BB63.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C03BB64.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C03BB65.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

% EMG\_Names(i) = 'N2001C03BB66.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C03BB67.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C03BB68.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C03BB69.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C03BB70.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C03BB71.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C03BB72.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C03BB73.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C03BB74.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C03BB75.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C03BB76.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C03BB77.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C03BB78.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C03BB79.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

EMG\_Names(i) = 'N2001C04BB51.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C04BB52.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C04BB53.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C04BB54.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C04BB55.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C04BB56.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C04BB57.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C04BB58.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C04BB59.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C04BB60.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C04BB61.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C04BB62.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C04BB63.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C04BB64.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C04BB65.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

% EMG\_Names(i) = 'N2001C04BB66.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C04BB67.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C04BB68.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C04BB69.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C04BB70.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C04BB71.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C04BB72.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C04BB73.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C04BB74.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C04BB75.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C04BB76.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C04BB77.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C04BB78.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C04BB79.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

EMG\_Names(i) = 'N2001C05BB51.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C05BB52.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C05BB53.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C05BB54.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C05BB55.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C05BB56.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C05BB57.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C05BB58.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C05BB59.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C05BB60.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C05BB61.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C05BB62.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C05BB63.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C05BB64.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C05BB65.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

% EMG\_Names(i) = 'N2001C05BB66.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C05BB67.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C05BB68.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C05BB69.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C05BB70.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C05BB71.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C05BB72.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C05BB73.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C05BB74.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C05BB75.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C05BB76.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C05BB77.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C05BB78.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C05BB79.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

EMG\_Names(i) = 'N2001C06BB51.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C06BB52.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C06BB53.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C06BB54.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C06BB55.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C06BB56.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C06BB57.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C06BB58.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C06BB59.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C06BB60.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C06BB61.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C06BB62.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C06BB63.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C06BB64.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C06BB65.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

% EMG\_Names(i) = 'N2001C06BB66.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C06BB67.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C06BB68.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C06BB69.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C06BB70.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C06BB71.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C06BB72.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C06BB73.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C06BB74.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C06BB75.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C06BB76.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C06BB77.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C06BB78.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C06BB79.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

EMG\_Names(i) = 'N2001C07BB51.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C07BB52.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C07BB53.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C07BB54.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C07BB55.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C07BB56.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C07BB57.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C07BB58.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C07BB59.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C07BB60.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C07BB61.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C07BB62.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C07BB63.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C07BB64.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

% EMG\_Names(i) = 'N2001C07BB66.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C07BB67.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C07BB68.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C07BB69.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C07BB70.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C07BB71.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C07BB72.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C07BB73.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C07BB74.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C07BB75.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C07BB76.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C07BB77.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C07BB78.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C07BB79.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

EMG\_Names(i) = 'N2001C08BB51.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C08BB52.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C08BB53.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C08BB54.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C08BB55.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C08BB56.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C08BB57.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C08BB58.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C08BB59.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C08BB60.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C08BB61.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C08BB62.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C08BB63.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C08BB64.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

% EMG\_Names(i) = 'N2001C08BB65.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C08BB66.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C08BB67.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C08BB68.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C08BB69.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C08BB70.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C08BB71.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C08BB72.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C08BB73.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C08BB74.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C08BB75.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C08BB76.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C08BB77.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C08BB78.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C08BB79.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

EMG\_Names(i) = 'N2001C09BB51.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C09BB52.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C09BB53.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C09BB54.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C09BB55.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C09BB56.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C09BB57.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C09BB58.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C09BB59.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C09BB60.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C09BB61.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C09BB62.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C09BB63.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C09BB64.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

% EMG\_Names(i) = 'N2001C09BB65.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C09BB66.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C09BB67.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C09BB68.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C09BB69.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C09BB70.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C09BB71.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C09BB72.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C09BB73.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C09BB74.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C09BB75.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C09BB76.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C09BB77.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C09BB78.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C09BB79.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

EMG\_Names(i) = 'N2001C10BB51.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C10BB52.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C10BB53.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C10BB54.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C10BB55.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C10BB56.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C10BB57.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C10BB58.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C10BB59.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C10BB60.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C10BB61.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C10BB62.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001C10BB63.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

% EMG\_Names(i) = 'N2001C10BB64.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C10BB65.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C10BB66.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C10BB67.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C10BB68.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C10BB69.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C10BB70.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C10BB71.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C10BB72.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C10BB73.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C10BB74.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C10BB75.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C10BB76.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C10BB77.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C10BB78.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

% EMG\_Names(i) = 'N2001C10BB79.bin';

% [EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

% i = i + 1;

sizeCTRL = i - sizeALS;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% LOAD MYOPATHY DATA

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

EMG\_Names(i) = 'N2001M02BB01.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M02BB02.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M02BB03.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M02BB04.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M02BB05.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M02BB06.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M02BB07.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M02BB08.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M02BB09.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M02BB10.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M03BB01.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M03BB02.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M03BB03.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M03BB04.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M03BB05.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M03BB06.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M03BB07.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M03BB08.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M03BB09.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M03BB10.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M03BB11.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M03BB12.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M03BB13.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M03BB14.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M03BB15.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M04BB51.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M04BB52.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M04BB53.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M04BB54.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M04BB55.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M04BB56.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M04BB57.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M04BB58.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M04BB59.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M04BB60.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M04BB61.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M05BB51.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M05BB52.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M05BB53.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M05BB54.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M05BB55.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M05BB56.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M05BB57.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M05BB58.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M05BB59.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M05BB60.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M05BB61.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M05BB62.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M05BB63.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M05BB64.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M05BB65.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M06BB01.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M06BB02.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M06BB03.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M06BB04.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M06BB05.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M06BB06.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M06BB07.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M06BB08.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M06BB09.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M06BB10.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M06BB11.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M06BB12.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M06BB13.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M06BB14.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M06BB15.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M07BB01.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M07BB02.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M07BB03.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M07BB04.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M07BB05.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M07BB06.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M07BB07.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M07BB08.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M07BB09.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M07BB10.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M07BB11.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M07BB12.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M07BB13.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M07BB14.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M07BB15.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M07BB16.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M07BB17.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M07BB18.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M07BB19.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M07BB20.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

i = i + 1;

EMG\_Names(i) = 'N2001M07BB21.bin';

[EMG\_Master\_TimeData(:,i), EMG\_Master\_PSDData(:,i)] = uploadEMG(EMG\_Names(i));

sizeMYO = i - sizeCTRL - sizeALS;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Create Targets and Other Potential Inputs

% Autoregression (AR), root mean squared (RMS),

% variance,

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

EMG\_AR = aryule(EMG\_Master\_TimeData, 7);

EMG\_AR = transpose(EMG\_AR);

EMG\_medfreq = medfreq(EMG\_Master\_PSDData);

EMG\_varVAL = var(EMG\_Master\_TimeData);

EMG\_rmsVAL = rms(EMG\_Master\_TimeData);

LearnResponses = ones(3, i).\* 0;

for j = 1:i

if(j <= sizeALS)

LearnResponses(3, j) = 1;

elseif(j <= (sizeALS + sizeCTRL))

LearnResponses(1, j) = 1;

else

LearnResponses(2, j) = 1;

end

end

**B2. Upload EMG Data Custom Function**

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Course : EGR 434

% Instructor : Dr. Rhodes

% Author : Nicolas Fernandez and Jordan Hayes

% Assignment: ALS Detection Final Project

% Description: A custom function that takes

% an EMG .bin file name as a string argument

% and stores the time domain signal as well as

% the Power Spectral Density using the discrete

% Fast Fourier Transform of the signal.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [x, PSD] = uploadEMG(filename)

fileID = fopen(filename);

x = fread(fileID);

fclose(fileID);

n = length(x);

PSD = abs(fft(x, n)).^2;

**B3. Optimal Network Code (MATLAB Generated)**

% Solve a Pattern Recognition Problem with a Neural Network

% Script generated by Neural Pattern Recognition app

% Created 07-Dec-2021 23:44:39

%

% This script assumes these variables are defined:

%

% EMG\_AR - input data.

% LearnResponses - target data.

x = EMG\_AR;

t = LearnResponses;

% Choose a Training Function

% For a list of all training functions type: help nntrain

% 'trainlm' is usually fastest.

% 'trainbr' takes longer but may be better for challenging problems.

% 'trainscg' uses less memory. Suitable in low memory situations.

trainFcn = 'trainscg'; % Scaled conjugate gradient backpropagation.

% Create a Pattern Recognition Network

hiddenLayerSize = 20;

net = patternnet(hiddenLayerSize, trainFcn);

% Choose Input and Output Pre/Post-Processing Functions

% For a list of all processing functions type: help nnprocess

net.input.processFcns = {'removeconstantrows','mapminmax'};

% Setup Division of Data for Training, Validation, Testing

% For a list of all data division functions type: help nndivision

net.divideFcn = 'dividerand'; % Divide data randomly

net.divideMode = 'sample'; % Divide up every sample

net.divideParam.trainRatio = 70/100;

net.divideParam.valRatio = 15/100;

net.divideParam.testRatio = 15/100;

% Choose a Performance Function

% For a list of all performance functions type: help nnperformance

net.performFcn = 'crossentropy'; % Cross-Entropy

% Choose Plot Functions

% For a list of all plot functions type: help nnplot

net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...

'plotconfusion', 'plotroc'};

% Train the Network

[net,tr] = train(net,x,t);

% Test the Network

y = net(x);

e = gsubtract(t,y);

performance = perform(net,t,y)

tind = vec2ind(t);

yind = vec2ind(y);

percentErrors = sum(tind ~= yind)/numel(tind);

% Recalculate Training, Validation and Test Performance

trainTargets = t .\* tr.trainMask{1};

valTargets = t .\* tr.valMask{1};

testTargets = t .\* tr.testMask{1};

trainPerformance = perform(net,trainTargets,y)

valPerformance = perform(net,valTargets,y)

testPerformance = perform(net,testTargets,y)

% View the Network

view(net)

% Plots

% Uncomment these lines to enable various plots.

%figure, plotperform(tr)

%figure, plottrainstate(tr)

%figure, ploterrhist(e)

%figure, plotconfusion(t,y)

%figure, plotroc(t,y)

% Deployment

% Change the (false) values to (true) to enable the following code blocks.

% See the help for each generation function for more information.

if (false)

% Generate MATLAB function for neural network for application

% deployment in MATLAB scripts or with MATLAB Compiler and Builder

% tools, or simply to examine the calculations your trained neural

% network performs.

genFunction(net,'myNeuralNetworkFunction');

y = myNeuralNetworkFunction(x);

end

if (false)

% Generate a matrix-only MATLAB function for neural network code

% generation with MATLAB Coder tools.

genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes');

y = myNeuralNetworkFunction(x);

end

if (false)

% Generate a Simulink diagram for simulation or deployment with.

% Simulink Coder tools.

gensim(net);

end