Robust EMG Classification Using Sparse Recovery

17 May 2018

Peter Klinkmueller
pklinkmueller@jhu.edu

Harrison Nguyen
hnguye68@jhu.edu

Aurik Sarker
asarker1@jhu.edu

Akhil Vasvani
avasvan1@jhu.edu

***Abstract*—This paper examines the merits of modifying sparse recovery algorithms in order to improve classification of limb positions from electromyography (EMG) signals. In particular, we investigate performance in cases where the signal has degraded due to electrode shift and noise, causing classification accuracy to break down. Sparse Representation Classification (SRC) offers a promising approach for robust EMG classification with prior success in other noise-prone classification problems. We hope to demonstrate how certain dictionary-manipulating methods may enhance already existing recovery techniques in both improving accuracy in these cases and reducing computation time. Specifically, we shall explore dictionary augmentation, basis projection, dictionary compression, and sparse voting schemes.**

***Index Terms*—electromyography, myoband, prosthesis, limb position, electrode shift, sparse recovery, SRC, ELM, EASRC, dictionary learning, time series, machine learning**

MOTIVATION

Electromyography, or EMG, is a technique for analyzing myoelectric signals resulting from electrical activity in the muscle during a contraction. EMG is recorded by placing surface electrodes on the skin to measure electric potentials from muscles while a limb contracts or moves. These signals can be classified to decode an actual or intended limb position, offering amputees a noninvasive means of controlling a prosthesis as shown in Fig. 1. This control can be implemented with pattern recognition using traditional machine learning algorithms for classification, such as LDA or SVM, utilizing an EMG electrode array in real time [1-7].

However, in attempting real-time classification of limb positions we encounter a key issue with the electrodes in the EMG electrode array. In practice, we observe a phenomenon called ”electrode shift”, which occurs when the electrodes move either radially or laterally along the arm. In this case, the classification will not be as accurate, as the myoelectric signals vary based on the actual positions of the electrodes. Since the EMG electrodes are measuring different groups of muscle fibers than what was recorded when training a given classifier, amputees observe a noticeable decrease in classification accuracy, down to roughly 68% for SRC and 59% percent for LDA.

To address this, we look toward sparse recovery methods instead. Sparse Recovery Classification, or SRC, takes advantage of the fact that our recovered vector (indicating which class was identified) is sparse [8] [9]. This method is quite robust to signal degradation, and will not lose much accuracy in the presence of noise that would break down the classification ability of other methods. However, it too suffers from great computational complexity requirements and does not always produce good results. In this paper, we hope to build on this algorithm, specifically by modifying the training dictionary in various ways in an effort to reduce both this complexity and its error rate due to EMG electrode shifting.

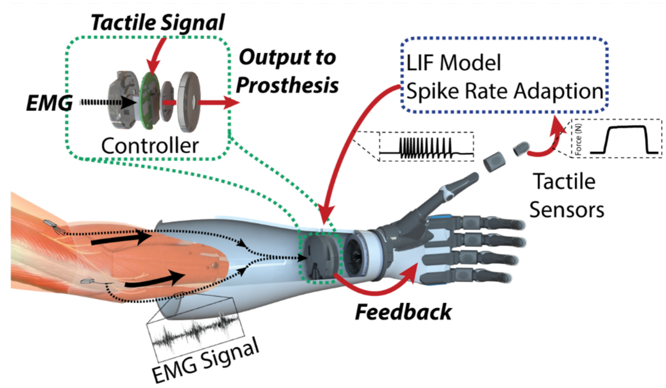


Fig. 1. **General schematic for prosthesis control. EMG signals recorded from the amputee are fed into the controller and classified as a certain pose or gesture, outputting a control signal to the prosthesis to actuate its fingers in accordance to the classification label.**

I. CURRENT METHODS

The library of machine learning techniques for classification is already quite extensive, as the academic community has already devoted large efforts in this domain. We introduce here only a couple of these techniques popular among researchers in prosthesis control, particularly due to their ease of implementation and low computational complexity. Generally, supervised learning methods are used since the poses or motions associated with a recorded set of EMG signal patterns are imaginable, and in some cases even sensed by a phenomena referred to as ”Phantom Limb” sensation, by an amputee and are observable by an able-bodied subject. Therefore the EMG signals recorded during training are labeled.

A. Linear Discriminant Analysis

Clustering can be quite difficult in circumstances where there is not a clear distinction between classes in higher-dimensional space. Linear Discriminant Analysis, or LDA, attempts to alleviate this issue by determining a subspace projection which maximizes the distance between data clusters while minimizing the variance within individual clusters. This newfound separation makes it easier to classify the clusters later on. To do so, there are two routes to be considered: within class separation and between class separation. The equations are:

$$S_w = \sum_{i=1}^{c} \sum_{x \in D_i}^{n} (x - m_i)(x - m_i)^T$$

$$S_b = \sum_{i=1}^{c} N_i (m_i - m)(m_i - m)^T$$

The first equation is for within class separation where S_w is the within class scatter matrix m_i is the mean vector. For the second equation, S_b is the between class matrix, m is the overall mean, m_i is the sample mean, and N_i is the size of the respective classes.

One of the earliest EMG control strategies investigated using LDA on time domain features [2]. In the EMG problem defined above, Joseph Betthauser et al experimented with using LDA to improve classification [7].

While LDA improves the accuracy of the classifier from the baseline, it provides only a decent level of robustness to electrode shift. Additionally, LDA requires extra computation time in searching for the right transformation.

B. Support Vector Machine

One very effective method of performing binary classification, i.e. clustering of two classes, is the support vector machine. Support Vector Machine, or SVM, is a discriminative classifier which attempts to determine the optimal hyperplane which separates two classes given from training data. It can also t can also This trained discriminator can then be used classify incoming test data extremely quickly, making it optimal for use in real-time prosthesis control. As with LDA, however, the SVM classifier is unable to provide robustness against the issue of electrode shift.

C. Extreme Adaptive Sparse Recovery Classification

Earlier, we introduced Sparse Recovery Classification (SRC) as a more robust method of classifying data. It works by solving the optimization problem outlined in the following problem statement:

$$\hat{\mathbf{x}}_1 = argmin\|x\|_1 \quad \text{subject to} \quad \mathbf{Ax} = \mathbf{y}$$

taking advantage of the sparsity of the classification vector [8] [9]. Because this $l1$-minimization problem searches for solutions which are sparse, it is able to disregard solutions which include incorrect values from other classes. Thus, SRC acts as a very robust classification method to various types of noise.

To solve this optimization problem, we can employ a variety of different algorithms. One very popular method is Orthogonal Matching Pursuit, which iteratively finds values of the recovered vector, as well as its support. Unfortunately, this algorithm, and other efficient SRC solvers, are still quite slow, and do not scale well as the dictionary increases in size.

To alleviate this problem, the field of EMG classification has instead explored the use of an enhanced version of SRC, called Extreme Adaptive Sparse Recovery Classification, or EASRC [10]. EASRC seeks to provide robust classification with very little computational cost by combining the best features of SRC together with a type of shallow, feed forward neural network called an Extreme Learning Machine (ELM). EASRC works as a cascading algorithmic solver, utilizing the best features of the ELM and SRC algorithms it is based upon. For most test cases, EASRC uses ELM, which is less computationally complex than SRC, but also more inherently sensitive to noise (highly detrimental to the specific problems we are addressing). However, if the ELM fails to classify a test vector at an adequate confidence level, the EASRC sends this vector to the SRC algorithm for classification. This hybrid method runs orders of magnitude faster than traditional SRC, while still maintaining similar accuracy due to the ELM's use of a confidence metric for deciding when to rely on SRC for robust classification of particularly difficult test cases. Prior work has been done in using EASRC as a robust method for myoelectric control in the context of limb position classification for amputee prosthesis control [6] [7].

II. Our Methods

As biosignals, such as EMG, are non-stationary due to their nature, e.g. action potentials resulting from the flux of ions across neuron membranes are stochastic in nature, signal processing methods are required to extract meaningful features from an EMG signal and multiple recordings of EMG signals for each muscle contraction must be recorded to identify meaningful patterns via pattern recognition or machine learning. However, since training each desired pose or motion for a classification algorithm creates burden on the subject by contracting the relevant muscles for each pose, the observations available from the training data is limited. If we measure both time domain and frequency domain features, then the dimensionality of the classification problem is often greater than the number of observations available. This creates a tradeoff between using a rich amount of information to distinguish each pose and having reliable classification. Especially for classification algorithms based on sparse recovery, having a "tall" training dictionary increases computation time as well. Thus, we aim to modify the training dictionary to increase computation speed and/or increase classification performance.

A. Signal Processing

To compress the amount of information available in each EMG signal recording, common signal processing methods for EMG signal analysis are used. Since EMG is based on the electric potentials generated from muscles contracting at a frequency range of (0-450 Hz), EMG signal analysis lends itself to examining time domain features and frequency features simultaneously. Both of these sets of features are used to examine the magnitude and frequency of contractions. To avoid aliasing the sampled frequencies from anatomical contractions, the EMG signal is sampled at $1000Hz$. To segment the data, the EMG signal obtained from each pose or motion is segmented and analyzed in $200ms$ windows. Once time domain and frequency domain features are extracted from the 8 channels, each feature set is concatenated into a single vector. For each collection of signals, the vectors are concatenated into a dictionary that is feature scaled, where each feature is normalized to a range of $[0, 1]$ by subtracting the minimum value of each feature and dividing by the maximum value of each feature. Feature scaling improves classification performance since the range of each time domain feature and frequency domain features vary significantly in magnitude.

1) Time domain: Within a given window, simple time domain features from EMG can be used as markers for the strength and frequency of a contraction [1] [2] [6]. These features generally only involve elementary math operations such as summing and division, which allows for efficient and flexible use, as well as serving as a dense quantifiable metric to sample. Sparse recovery-based classification methods in particular take advantage of densely sampled data.

- Mean Absolute Value (MAV) is computed by averaging the absolute value of the EMG signal. MAV is proportional to contraction strength since more muscles are recruited, generating more action potentials which are sensed by the EMG electrodes.
- Variance (VAR) is computed for the EMG amplitude. VAR is another correlate of EMG signal strength from contraction.
- Wave Length (WL) is found by summing up the absolute value of the EMG signal slope, between two consecutive time samples, over the window. WL increases with both EMG amplitude and frequency increases as contraction strength increases.
- Zero Crossings (ZC) is found by summing up the number of instances where the EMG amplitude crosses $0mV$ and the change is above a certain threshold, which depends on the subject but typically is on the order of 10^{-6} to filter out noise, within the window. ZC is linked to muscle activation since changes in EMG frequency are influenced by weaker and deeper muscles. One can estimate frequency by dividing ZC by double the window length and multiplying it by the number of windows per second.
- Sign Slope Changes (SSC) is found by summing up the number of instances where the slope changes sign and the change is greater than a threshold, which varies from subject to subject like zero crossings but is on a similar order of magnitude of 10^{-6}, within the window. SSC is another correlate for muscle activation.

2) Frequency domain: To analyze frequency content of EMG, a 128 length FFT is computed. Since the FFT is symmetric however, only 64 components are “useful” in containing unique information. Frequency content of EMG is useful since muscles are comprised of different fiber types which contract at different frequencies depending on the type of muscle contraction being performed such as gross vs. fine movement [1] [6].

B. Dictionary Augmentation

One approach to making the training dictionary ”fat” is by performing recovery on an augmented dictionary using both time domain and frequency domain features. The training data dictionary is augmented by the identity matrix. The A matrix is a 1064 x T matrix, so we wish to augment the matrix by a 1064 x 1064 dimension identity matrix. The recovery problem therefore is formulated as:

$$y = Ax + \varepsilon \rightarrow y = \begin{bmatrix} A & I \end{bmatrix} \begin{bmatrix} x \\ \varepsilon \end{bmatrix}$$

Based on lecture, we aim to increase classification accuracy by attempting to encode the ”error” in the measurement vector. Specifically, by augmenting the training dictionary with the identity matrix, the sparse code we recover will have both the sparse code of the training dictionary and the code for the error. The goal is to recover the first T samples after the augmentation.

C. Basis Projection

Another method we attempted based on the course material was a basis projection of the problem. A basis projection is a simple method that attempts to reduce the mutual coherence of the the dictionary, thereby increasing the potential for SRC to be able to discriminate efficiently between the columns (entries) it contains. The coherence of our dictionary is defined as:

$$\mu(\mathbf{A}) = \max_{1 \le i \ne j \le N} |\langle \mathbf{a}_i, \mathbf{a}_j \rangle|$$

- the largest absolute inner product between any two normalized columns $(\mathbf{a}_i, \mathbf{a}_j)$. From class, we were able to see that incoherence implies the RIP (Restricted Isometry Property) holds, which in turn implies the NUP (Null Space Property) holds, which, finally, is materially equivalent to recovery success. Our hope in attempting this modification of traditional SRC is to improve the accuracy of our algorithm for the problem of pose classification by providing a more incoherent dictionary to result in better sparse representations of test signals. The better the sparse representations, the more robust the classification via our algorithm becomes. The basic mathematical idea of basis projection can be seen as transforming the equation that we endeavor to solve for the recovery problem:

$$y = Ax \Rightarrow Ry = RAx$$

Where the R matrix in question is our projection. We attempt use of random normal, random uniform, and DCT (Discrete Cosine Transform) matrices for our projection matrix. The modification to SRC is applied within the EASRC framework, meaning that it will only have an effect when the SRC portion of the algorithm is utilized for classification in the more difficult test cases the EASRC method faces. Due to this, we do not expect to see leaps and bounds of improvement (if the augmentation only operates on a subset of test cases, it cannot dramatically impact the overall classification accuracy), but instead look to see if the projection can aid the robust classification of the more challenging test cases when SRC is used.

D. Dictionary Compression

As mentioned earlier, computational complexity and runtime (specifically, the ability to run *online* classification in about 20 milliseconds) are crucial to the practical success of our algorithm. Therefore, it may be beneficial to reduce the size of the dictionary, as computation time for the sparse recovery algorithm improves dramatically with even a modest amount of dictionary compression. However, this dictionary compression must be done in such a way that does not drastically change the span of each class in N-dimensional space to avoid great reduction in the classification accuracy. The figure below demonstrates compression in the N dimension.

$$A = \begin{bmatrix} | & | & & | \\ a_1 & a_2 & \dots & a_N \\ | & | & & | \end{bmatrix} \rightarrow \begin{bmatrix} | & | & & | \\ a_1 & a_2 & \dots & a_{N'} \\ | & | & & | \end{bmatrix}$$

1) Random Subsampling: The simplest way of going about this is random subsampling of the training vectors, reducing the size of the dictionary in the N dimension. This simply entails selecting a random subset of the training vectors and testing on the resulting dictionary. Because the original dictionary was already quite extensive (containing many data point for each class), it is unlikely that removal of a subset would drastically effect the classification of any particular class by much. Because of this assumption, we can reduce the dictionary drastically and not observe a very large reduction in classification accuracy. While this compression method is easy to implement and computationally simple, it does lead to some variability in the results due to the random nature of the process.

2) K-Means and K-Medians: In order to avoid the aforementioned issue with randomness, we turn to a deterministic algorithm for dictionary compression. Here, each class is reduced independently, so that we end up having the same number of training vectors for each of them (k). This is done so that data vectors do not get mislabeled during k-means. This method takes advantage of redundancy within classes in the training data; this redundancy allows us to be able to represent each class with fewer data points. It may also remove certain outlier training points which are not representative of the class as a whole.

Depending on the shape of the class clusters, however, this k-means algorithm may learn means which are not even part of the original dataset. In this case, it may be beneficial to use k-medians, which represents clusters using a specific, real datapoint from the class. The implementation is almost identical, except that the clusters are updating using the median vector from the training set instead of the mean.

3) Dimensionality Reduction: Another way to decrease the size of the training dictionary is to reduce the dimensionality of the individual training vectors; in other words, reduce the dictionary size in the M dimension. This form of compression has the added benefit of making the training matrix more ”fat”, or increases the ratio between the M and N dimension, which improves the performance of the SRC algorithm. One way of doing so is called Principal Component Analysis, or PCA, which attempts to find a lower-dimensional eigenspace of weights to represent each data point.

However, performing PCA runs the risk of losing important information in certain dimensions. In this case, we may use what we know about the frequency content of these EMG signals to remove dimensions which we know are not relevant. Although the Fourier coefficients give us signals are sampled at 1000 Hz, the information about limb orientation mostly resides in the frequencies between around 20 and 250 Hz. This implies that we may crop the dictionary along that dimension without losing information about limb position.

E. SRC Voting

Inspired by the occlusion problem formulated for face recognition [8] , we aim to split the training dictionary into subdictionaries for each feature and perform multiple recoveries. The resulting class each feature is predicted to come from is treated as a vote from each feature. The recovery problem is now formated as follows:

$$y = Ax \rightarrow \begin{bmatrix} y_{td_1} \\ \vdots \\ y_{td_5} \\ y_{ft_1} \\ \vdots \\ y_{ft_4} \end{bmatrix} = \begin{bmatrix} A_{td_{11}} & \dots & A_{td_{1C}} \\ \vdots & \dots & \vdots \\ A_{td_{51}} & \dots & A_{td_{5C}} \\ A_{ft_{11}} & \dots & A_{ft_{1C}} \\ \vdots & \dots & \vdots \\ A_{ft_{41}} & \dots & A_{ft_{4C}} \end{bmatrix} \begin{bmatrix} x_{td_1} \\ \vdots \\ x_{td_5} \\ x_{ft_1} \\ \vdots \\ x_{ft_4} \end{bmatrix}$$

$$\begin{aligned} y_{td_1} &= A_{td_1} x_{td_1} \\ &\vdots \\ y_{td_5} &= A_{td_5} x_{td_5} \\ y_{ft_1} &= A_{ft_1} x_{ft_1} \\ &\vdots \\ y_{ft_4} &= A_{ft_4} x_{ft_4} \end{aligned}$$

Where each of the td_i and ft_j are time domain and frequency domain features, respectively. The time domain features are simply the five domain features described earlier, and the frequency domain features are uniform length subbands within the FFT computed for a single window. Four frequency subbands were found to provide the best results, leading to nine total recovery problems being performed before the final classification. We can attempt this for the original SRC algorithm and the SRC branch of EASRC.

III. Experimental Setup

The MiniVIE package, vMPL GUI, and Myoband (utilized for demo of practical, technological potential) were used for our project implementation.

The work presented in this project is based off of a data set provided by the Johns Hopkins Neuroengineering and Biomedical Instrumentation Laboratory. The Thakor lab experiment, from which the data for this project was derived utilized a 3 able-bodied, 3 amputee subject setup. The data set consists of an experiment that quantifies how electrode shift affects EMG classification performance for 9 poses. Each subject wore a custom EMG electrode array with 8 channels sampled at 1000 Hz. At the start of the experiment, each subject was asked to elicit a movement or contraction corresponding to 9 different poses (rest as a baseline class, hand open & hand close, wrist pronation & wrist supination, wrist flexion & wrist extension, point & key pose) each for a few minutes each. Once all 9 poses were trained, the electrode array was then shifted 1 cm radially and laterally in 1 cm increments between each training session. The max displacement corresponded to 3 cm from the initial position since any electrode shift corresponding to 3 cm or more, from empirical evidence, suggests that the electrode array has poor and unreliable contact. In total, 9 sets of data were collected from each coordinate on a 3 cm x 3 cm grid. Since a window of 200 ms is analyzed as a training sample and each pose is trained for a few minutes each, the training dictionaries would have roughly 400 samples compared to a feature space of 1064. The data is limited to this estimate, since repeating each pose for several minutes for each electrode position creates a burden on the subject.

For the classification task, one set of classification data is chosen as the training set. The 8 other sets of classification data can be treated as a collection of test vectors or measurements "from 8 different sources" for a given training set. Since there are 9 sets of data, this allows for 72 possible testing scenarios for how the electrodes can shift given the constraint of 1 cm translation on a 3 cm x 3 cm grid.

IV. Results

A. Dictionary Augmentation

This method was initially explored using a single subject's dataset. Even though, in the results shown above, the method implemented (SRCv) performs slightly better than the unmodified SRC method, the improvement is not of great enough magnitude to merit further exploration. In addition, the computation time to perform this was excessive (multiple hours for a single subject). Hence, we deemed there to be insufficient motivation to apply this method to all subjects due to the computational complexity and limited improvement seen from the first subject. One future implementation to potentially reduce the computation time would be to subsample the identity matrix or zero out rows of it. Instead of recovering x, we wish to cluster rows of the augmented dictionary. This could reduce run time and

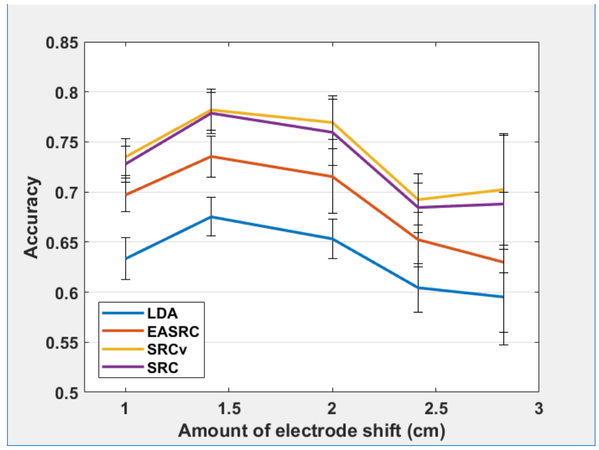


Fig. 2. **Results for Dictionary Augmentation versus LDA, SRC, and EASRC.**

accelerate our process. However, one drawback might be increasing the error. Nonetheless, it would yield an interesting comparison between the two. Additionally, we declined to investigate this method further as the realization was had that it was unlikely we were dealing with significant noisiness of the data in a form extractable by this type of augmentation. Rather, the shifting of the EMG arrays is more probable to be manifested in a far more deliberate manner than simple added noise. Thus, we found this dictionary augmentation to be not of use for this application.

B. Basis Projection

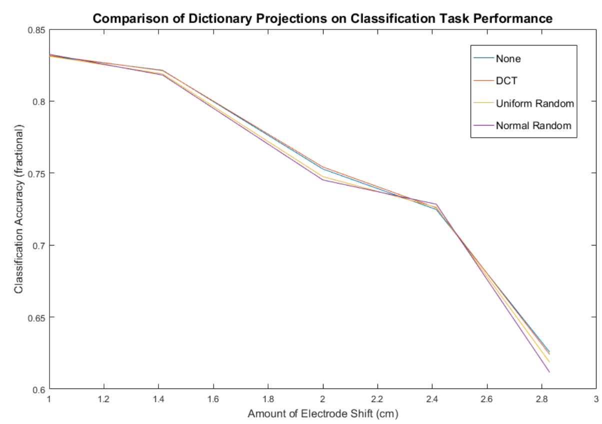
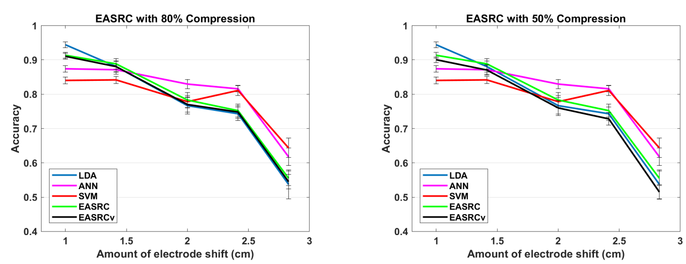


Fig. 3. **Results for EASRC classification with different SRC projections.**

We did not see any significant improvement from utilization of any of the tested basis projections applied to the SRC portion of the EASRC classification algorithm. While some of the projections showed promise within a number of subsets of the data (most significantly, the random normal projection), none of the projections seemed to add measurable robustness to the EASRC classification. The success or failure of the projections to be effective at increasing the discriminatory ability of the SRC algorithm may be partially obscured by the fact that the fraction of test cases handled by the SRC portion of EASRC is very small, so any impact would be limited. However, we would have hoped that in these very cases when SRC is applied, on the most difficult test cases the EASRC algorithm sees, the projection might be able to lend additional robustness. Instead, it appears that the various projections offer little across the board in terms of improved classification. The runtime of classification with the added projections was largely equivalent, but did see a very slight rise over the unmodified EASRC runtime in the case of the inverse DCT projection, since a matrix inversion operation was performed for each test case handled by the SRC portion of the algorithm in that case. In the random normal and random uniform projection cases, the runtime effect was expectedly negligible. While it is possible that a projection exists that could increase SRC classification robustness, we were unsuccessful in finding one (note: while not shown in this plot, the Bernoulli random projection was also explored, but ultimately did not show enough promise to fully test).

C. Dictionary Compression

This method was tested on a single subject, with only features in the time domain, as using vectors of the size of the frequency coefficients took much longer to compute (when performing compression) while achieving similar results to those in the time domain. The results for running on this single subject using either 80% or 50% compression are shown below, in the figure below.

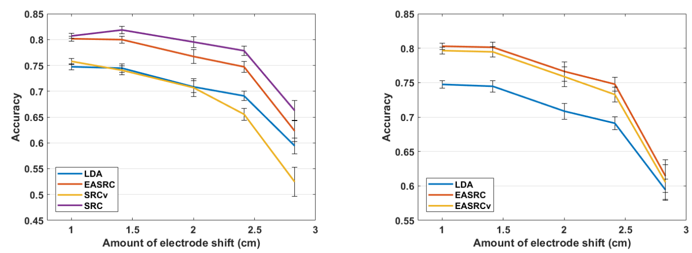


(a) 80% Compression (b) 50% Compression

Fig. 4. **Results for EASRC classification with different levels of compression**

Here, we see similar accuracies to the results both with and without compression. This is to be expected, as removing vectors from the training dictionary is equivalent to removing information which could have been useful in classifying certain outlier test points. What is promising about these results is that we retain the accuracy of the original EASRC algorithm without compression in both instances. This demonstrates what we hypothesized earlier, that removing vectors from the dictionary should not make much of an effect on the classification itself due to redundancy. Additionally, we found that this dictionary compression achieves its main goal, which was reduction of computation time. In the case where the training dictionary was reduced in size by half, the time required to perform one classification was 0.00028 seconds on average, compared to 0.00092 seconds for regular EASRC. This is a reduction in runtime almost double that of our reduction in training dictionary size

D. SRC Voting



(a) SRC Voting (b) EASRC Voting

Fig. 5. **Results for classification algorithms with multiple SRCs for each feature.**

Overall, voting appeared to have little benefit on the classification accuracy of the modified SRC (SRCv) and EASRC (EASRCv) algorithms. In terms of runtime, the computational complexity for multiple SRCs was noted to be significantly shorter than SRC on TD5 and FFT features, with some instances being faster by four-fold. However, we see that accuracy falls significantly at 2 cm of shift as compared to the other classification methods in Fig. 5a. For EASRC, EASRC with voting (EASRCv) actually increased run time up to 5-fold compared to the original implementation of EASRC. However, the voting scheme maintained comparable performance with EASRC as seen in Fig. 5b. More subjects would need to be analyzed before any definite conclusions can be made about the potential of this modification for increasing classification accuracy for our problem.

V. Conclusion

Overall, our modifications to SRC did not improve accuracy in any significant manner. This can be attributed to the robustness which we already have with SRC. However, the reductions in runtime which we noticed in performing either voting or dictionary compression were quite significant, and could allow for classification of limb position/orientation in real time. Given more time, parameters could be tuned to provide the same level of computation time improvement with better accuracy. Better information about the types of signals which shifted electrode arrays give us in relation to their original, non-shifted counterparts would allow us to better understand whether dictionary augmentation could really work, and perhaps identify a improved basis for these signals. Additionally, results seem to vary across different subjects, which explains the difference in accuracies across methods in our results sections

Future

For subsequent work, it would be beneficial to test this algorithm with a larger amount of data and improvements in the methods mentioned previously in the paper. For dictionary augmentation, we could subsample random multiple or different rows of the identity matrix. We can then extract further outputs of the electroshift to distinguish any significant improvements. For dictionary compression, using the information we know about the frequency vectors we can crop them and see if it results in any significant improvement in computation time. For SRC voting, instead of dividing up the dictionary into different types of features and different bins in the frequency axis. Instead, dividing the features by class and vote that way. This voting scheme would benefit from identifying which features are more reliable than others; in a similar vein, it may be more intelligent to split the frequency spectra by the contraction frequency range of different muscle fiber types. Also note that we can run combinations of these methods.

ACKNOWLEDGMENT

This research is supported by the Johns Hopkins Neuroengineering and Biomedical Instrumentation Laboratory, especially the guidance of Joseph Betthauser; none of this would have been possible without their prior work in the area. The prosthesis graphic is from Luke Osborn. We also thank Dr. Trac D. Tran, who supplied us with the requisite knowledge about sparse recovery to get us started on this project. It was a pleasure attending his course this semester.

REFERENCES

[1] B. Hudgins, P. Parker and R. N. Scott, "A new strategy for multifunction myoelectric control," in IEEE Transactions on Biomedical Engineering, vol. 40, no. 1, pp. 82-94, Jan. 1993.

[2] K. Englehart and B. Hudgins, "A robust, real-time control scheme for multifunction myoelectric control," in IEEE Transactions on Biomedical Engineering, vol. 50, no. 7, pp. 848-854, July 2003.

[3] A. J. Young, L. H. Smith, E. J. Rouse and L. J. Hargrove, "Classification of Simultaneous Movements Using Surface EMG Pattern Recognition," in IEEE Transactions on Biomedical Engineering, vol. 60, no. 5, pp. 1250-1258, May 2013.

[4] E. Scheme, K. Englehart, "Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use", J. Rehabil. Res. Devices, vol. 48, no. 6, pp. 643-659, 2011. bibitemscheme2 E. Scheme et al., "Selective classification for improved robustness of myoelectric control under nonideal conditions", IEEE Trans. Biomed. Eng., vol. 58, no. 6, pp. 1698-1705, Jun. 2011.

[5] C. Lin, B. Huang, N. Jiang, D. Farina, "Robust extraction of basis functions for simultaneous and proportional myoelectric control via sparse non-negative matrix factorization," J. Neural Eng., vol. 15, no. 2, Feb. 2018

[6] J. L. Betthauser et al., "Limb Position Tolerant Pattern Recognition for Myoelectric Prosthesis Control with Adaptive Sparse Representations From Extreme Learning," in IEEE Transactions on Biomedical Engineering, vol. 65, no. 4, pp. 770-778, April 2018.

[7] J. L. Betthauser, C. L. Hunt, L. E. Osborn, R. R. Kaliki and N. V. Thakor, "Limb-position robust classification of myoelectric signals for prosthesis control using sparse representations," 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Orlando, FL, 2016, pp. 6373-6376.

[8] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry and Y. Ma, "Robust Face Recognition via Sparse Representation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 2, pp. 210-227, Feb. 2009.

[9] Wei, Jiang-Shu, Jian-Cheng Lv, and Chun-Zhi Xie. "A New Sparse Representation Classifier (SRC) Based on Probability Judgement Rule." Information System and Artificial Intelligence (ISAI), 2016 International Conference on. IEEE, 2016.

[10] Cao, Jiuwen, et al. Extreme Learning Machine and Adaptive Sparse Representation for Image Classification. Neural Networks, vol. 81, 2016, pp. 91102., www.sciencedirect.com/science/article/pii/S0893608016300673.