Setting up Myo Armband

- Will update shortly

Acquiring data

- Will update shortly

Pre-Processing

- Rectified EMG, only 'abs' of Matlab was used

Working with NNMF

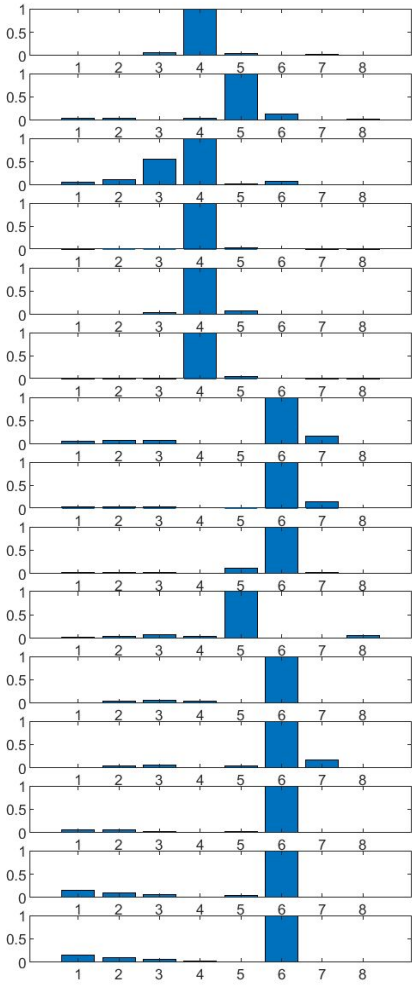
- There are p (total 20) participants and tr (15) trials by each participant in data
- X is a muscle data matrix created by adding all data from different hand movements.[8x72000]
    - X=[open hand, close hand, ……..]
    - X stands for a single participant and for a single trial
    - Therefore a 15x20 matric can be made which contain an X value for each cell
- To get synergies and activation commands, NNMF is applied to each X
- NNMF decomposed this data into W and H, based on the number of synergies we want to extract.
- No. of synergies (sy) was decided based on VAF and Rsq (Pearson correlation coefficient squared)
- 80% cutoff for VAF was considered
- X was decomposed 50 times X=>W*H, then W*H=>Xr
- VAF for all 50 Xr with respect to X was calculated
- Maximum VAF was considered to see how many synergies are required to recreate 80% or more data
- Based on this, 4 number of synergies was selected. Because 4 synergies can explain more than 80% VAF in our data
- Based on that max VAF for Xr was stored and corresponding W and H values were also saved
- Therefore now there is a 15*20 W matrix and a 15*20 H matrix available.
- Further when we would like to compare synergies 15 values of W matrix in a column will be compared sequentially
- tr=1:15, when tr=1 it gives us first trial for first participant (if p=1). W(tr=1,p=1) will be compared to W(tr=2,p=1) and likewise further upto tr reaches to 15.

Important knowledge about NNMF of Matlab-part1

- It was found out that the first column is having the highest Rsq if selected alone, then second.
- Third and fourth columns can get interchanged
- To know that W(:,i)*H(i,:)=Xrs was generated with all other values zero and compared as earlier Xr was compared to X
- Therefore Xrs vs X gives Rsq_single
- i=1-4 are first to 4th column, respectively

Comparision and Change in Sy structure

- **Method 1**
    - W(tr=1,p=1)=[sy1 sy2 sy3 sy4] was directly compared to W(tr=2,p=1)=[sy1 sy2 sy3 sy4]
    - 1-1,2-2,3-3,4-4 comparison using NDP
    - This was giving vague plots. Synergy structure which follows itself sometimes changes to other synergies



Plot for Sy1, P=5,tr=1:15

- **Method 2**
  - W(tr=1,p=1)=[sy1 sy2 sy3 sy4] was directly compared to W(tr=2,p=1)=[sy1 sy2 sy3 sy4]
  - 1-1,1-2,1-3,1-4, likewise 2-1,2-2,2-3,2-4 comparison using NDP

cNDP

W1sy1-w2sy1	W1sy2-W2sy1		

W1sy1-W2sy2	W1sy2-W2sy2		
W1sy1-W2sy3	W1sy2-W2sy3		
W1sy1-W2sy4	W1sy2-W2sy4		

[mcndp,m]=max(cNDP)

mcndp=>[max column1, max column2,max column3,max column4]

m=[loc of max in column]

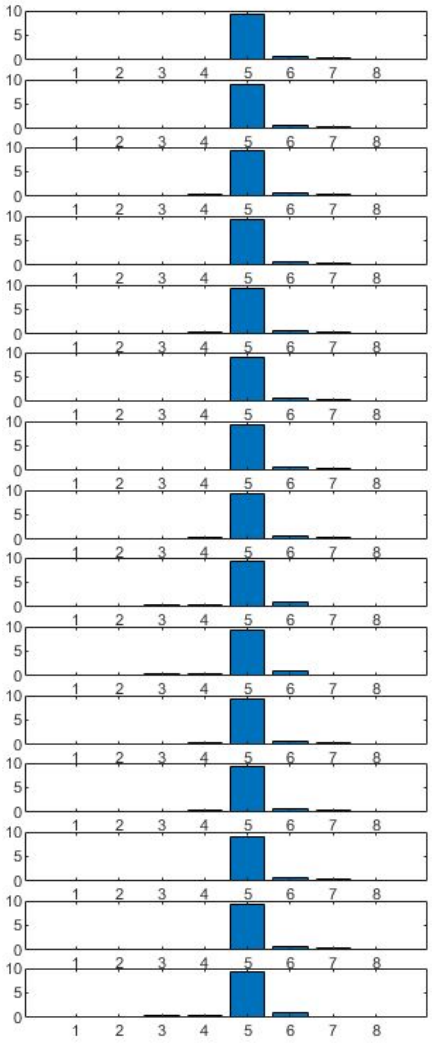
- If m=[1 2 3 4] it means NDP is highest for W1sy1-w2sy1, W1sy2-w2sy2, W1sy3-w2sy3, W1sy4-w2sy4
- If m=[2 1 4 3], it means NDP is high for (1-2,2-1,3-4,4-3) therefore we will change columns of W(tr=2,p=1) from 1-2-3-4 to 2-1-4-3
- This is done as per Valk's paper (2019) says that each sy should be compared to all other sy and the maximum normalized dot product can tell which synergy is related to which one and that's how the interchange of synergy columns should be there in the matrix
- Implementing this was sometimes troublesome when repeating values in m were showing up. for example m=[1 3 2 1]
- In that case, 1-1 and 4-1 are showing the highest NDP of the column. But if we implement that data of 4th column is replaced by 1 and data of 1st column is also replaced by 1. Therefore true representation of data wan not being generated after shuffling in columns
- To accommodate that NDP of those columns was compared which comes more than once in 'm' vector
- The highest NDP was given preference and likewise for the next one
- This was also problematic sometimes when multiple comparisons were required to be made
- This was not a reproducible result because W generated once was not same when we run the program next time
- To look for that change Dr. Bongers came up with an idea to compare the same data of X and recreated Xr 50 times and look for NDP between X and Xr(1:50)

Important knowledge about NNMF of Matlab-part 2

- I compared X and Xr and came on a conclusion that [mcndp,m]=max(cNDP) does not give m=[1 2 3 4] all the times
- This means that the same data can also get recreated in different ways keeping Rsq highest for 1st column of W and then 2nd, 3rd and then 4th (3,4 show nearby Rsq and are interchangeable because of a minute difference only)

- **Method 3**
  - I am not happy with this change in W(tr=2,p=1) recreation. 'm' should always remain [1 2 3 4] after recreation for the same synergy
  - Also, I am not happy with shuffling the synergy W(tr=2,p=1) with respect to W(tr=1,p=1)
  - To eliminate these two problems a new method is implemented
  - W(tr=1,p=1) was generated from X(tr=1,p=1) after iterating Xr 50 times and selecting max(VAF) of Xr. The corresponding W gives synergies for (tr=1,p=1)
  - For W(tr=2,p=1) we generate W 50 times from X(tr=2,p=1) and calculate NDP based on 2nd method (a 4*4 table was generated) [W(tr=1,p=1) vs W(tr=2,p=1)(for all 1:50)]
  - 'm' of that table was stored for all 50 combinations
  - Those 'm' were separated which give m=[1 2 3 4] out of 50 combinations
  - VAF of all these W(tr=2,p=1)*H=Xr for m=[1 2 3 4] was also saved simultaneously
  - max (VAF) => W(tr=2,p=1)*H=Xr, m=[1 2 3 4] is our final for W(tr=2,p=1)
  - This way 2nd trial synergy structure was selected
  - Likewise now W(tr=3,p=1) will come to the picture and so on all 15 trials will generate
  - An example image can be seen below
  - Now this has to be implemented to every trial and every participant
- Note:
  - I am worried about NNMF because it does not give an exact solution every time we generate W and H from X. Therefore, it will not be concrete to say how the synergies are changing.
  - Also, I am always working with the best data which gives m=[1 2 3 4]. But this data need not be the best match to the data the synergy structure has changed to.
  - Also, stacking the data based on NDP should not work always. Maybe due to learning firstly the change in synergy structure is less so NDP near to 1 is okay and max(NDP) works well there

- But what if the change in NDP in later trials becomes more i.e.NDP ~=0.6 then max(NDP) might not work there
- I am hopeful that method 3 will show the correct changes or maybe nearby solutions to exact solution because I am working with max(VAF for m= [1 2 3 4])



Plot for Sy1, P=1,tr=1:15

Methods

A: Ethical approval

B: Subjects

(copied from Franzke 2020)
Twenty able-bodied university students (mean age in years: 22(± 2.8), 11 females) were recruited. Handedness was assessed by completing the handedness questionnaire of the Edinburgh inventory [19], [20]. Exclusion criteria were any neurological pathologies or musculoskeletal complaints interfering with study outcomes.

C: Myoelectric machine learning system

Eight commercially available double differential electrodes were used (13E200=50 AC, Otto Bock Healthcare Products GmbH, Vienna, Austria), which pre-amplified and band-pass filtered the EMG signals. The electrodes were placed at equal distances around the thickest part of the forearm. EMG data were sampled at 1000 Hz and streamed to a laptop computer via Bluetooth connection. Software provided by Otto Bock Healthcare Products GmbH (Vienna, Austria) was used to record EMG data, to train a classifier, and to run a match-prompt test.

This data was recorded for another study in which a trial consisted of two blocks. In first block seven movements (eight with rest: wrist supination, wrist pronation, wrist flexion, wrist extension, hand open, fine pinch grip, and lateral thumb grip) were performed. This was done three times for the prompted levels of 30% maximum volumetric contraction (MVC), 60 % MVC and 90 % MVC. All of this recorded data was used to run a classifier to classify each movement in form of different EMG features. Later, the classifier was used to run a match prompt test in which online performance of the classifier was assessed.

D: Data utilization

The purpose of study performed in (Franzke 2020) was to assess the change in EMG feature metrics with user training. The system training utilized a linear discriminant analysis (LDA) classifier to train the model of muscle activities in terms of EMG features. Therefore, the user learning will be affected from the feature space and classifier selection.

We are using the same data, recorded for a LDA based training of 20 participants. This time we are utilizing a non-negative matrix factorization algorithm insead on each trial. It will give us

two lower dimensional subspaces named muscle synergies (W) and muscle activation coefficients (H). Each vector in these subspaces are considered as a feature.

E: Pre-processing

To apply NMF algorithm, the data matrix should be non-negative. For that 'abs' of Matlab R2018a was used.

The EMG muscle matrix data consists of three levels of MVC data. With 'rest' 24 activities make one data matrix. Therefore, the magnitude of data point is varying from zero to a very high value. To fit any model to such a dataset, normalization is required. Instead of normalizing with maximum amplitude of matrix (done in d'Avella 2003), the values in each row were normalized to the maximum value in that row, corresponding to the maximum level of muscle activity observed for that muscle across all conditions. Therefore, for each muscle all values ranged from 0 to 1. This was done to match the activity level in all electrodes. (done in Ting and Chatwal)

(more to try: 1. Before components are extracted using NMF, each muscle was also normalized to have unit variance, meaning that the sum of the squared values in the row equals 1. This allows the variations in each muscle to be considered with equal importance by the algorithm.(done in Ting and Chatwal)
2. If EMG onset cut will be required in data before and after each movement)

F: Synergy model

A fixed muscle synergy approach is adopted. In this approach it is assumed that the muscle synergies are fixed activation and do not vary with time. On the other hand muscle activation coefficient patterns are assumed to be time varying.

NMF algorithm was utilized to extract this information from the muscle activation data matrix. Each trial consists a matrix of dimension 8*72000. From this data 1 to 8 number of synergy structures can be extracted, which when linearly combined can explain for the actual recorded data. Therefore, it is essential that we choose a few ($N<8$) number of synergies which can recreate the actual data with a high accuracy.

Equation,

$$M = W*H+e$$

Is showcasing such a model, where W and H are the lower dimensional subspaces, muscle synergies and activation coefficients, respectively. Here, e denotes for the error in recreation or noise in other terms.

Therefore, for a high recreation the error (e) should be minimum. We assumed a 80% of recreation to be optimum to accept the synergy and activation coefficient matrices as a solution. The accuracy was calculated by highest variance accounted for (VAF). VAF can be given by equation

VAF= 1-(SSE/SST)

Here, SSE is the sum of squared errors of the data reconstructed by the muscle synergies (X=W*H) , and SST is the sum of squared residuals of data with respect to the mean of the different rows of the muscle data matrix (M).
VAF was calculated for each muscle data matrix 50 times. It was done to ensure that we find global minima of error between reconstructed X and main muscle matrix M. 1 to 8 synergies were extracted and based on highest VAF of each 1-8 synergies it was find out that 3 synergies can explain for approximately 70% of the data and 4 muscle synergies can explain for approximately 80% of the data. (approx because this figure changes if we redo the NMF)

#include figure of VAF vs number of Synergies

Therefore, it was decided that 4 muscle synergies will be explored in further analysis.

- **Working with function 'nnmf' of Matlab for NMF**
    - Function 'nnmf' of Matlab takes muscle data (M), and required muscle synergies (k) as input and provides W and H matrices as a solution. The maximum VAF based solution for W and H was calculated and stored for further analysis.
    - We want to track changes in muscle synergies and activation coefficients, therefore, we looked for the similarity in synergy structure in a pairwise manner. We are willing to perform three similarity analyses to see how the structure changes. These methods are normalized dot product (NDP), cosine of principal angles (CPA) and cluster analysis.
    - Similarity analysis NDP firstly of each trial to the next trial for every participant were calculated in a pairwise manner. It means that if each W has 4 muscle synergy vectors [W1 W2 W3 W4], W1 of first trial was compared to the W1 of next trial and so on. Similarly for W2, W3 and W4.
    - This did not worked well because it was found out theta the vectors W1.. W4 shuffles when we recalculate the solution again. Therefore, NDP of W1-W1

(adjacent trials) is not high always. It was found out that relation can be present in the solution but in a different shape. The data is shuffled in different columns and therefore to match correct columns we should not find NDP only in pairwise manner. We should find NDP of W1 to next W1-W4 (all) and should match with the next column which has highest NDP.

- Therefore, NDP was calculated in one to all combination manner. But it was found out that sometimes one column can have max NDP with two or more columns. This was problematic in allocating the next synergy a structure that is having high pairwise NDP. Also, if two columns have same data, it means one column has left some of the original data and therefore, this shuffled matrix will not carry the true information it should have.
- Further, we tried to find only those solutions for all Ws' which have W1-W1, W2-W2, W3-W3, W4-W4 maximum NDP, as well as we kept the constraint that the solution achieved is having at least 80% VAF. But this did not work out well because all trials were not able to produce a pairwise solution.
- Later it was also experienced that if the max VAF solution is changing every time, it means that the global minima of error is not getting achieved by the solution at all. Running 'nnmf' for a limited number of iterations will not be able to do that. Even if we will not be able to produce an exact and unique solution, our solution should be a nearby iteration which holds high similarity, atleast of 90% (personal claim, Valk 2019, Muceli 2014).
- Therefore, further I will be performing same operations with solutions achieved near the global minima. For that a customized NMF algorithm is being used for approx 10000-20000 iteration, and the criteria to solution is that the sum of all errors between X and Xr should be less then 10^-10 (i.e. sum(sum(X-Xr))<10^-10; one sum to add all columns, second to add all resultants of column sum; this value 10^-10 can be changed later dependent on accuracy we want to have in our solution and calculation time).
- There was a doubt about our data that if it is worthy to work with only 8 electrodes' data and if because of that the 'nnmf' or NMF in general can not produce an exact solution. It was found out that NMF in general can not produce an exact solution. The solutions will always be approximate and therefore to achieve global minima you should see how the error is being minimzed and at what point your error curve starts flattening up and that indicates the saturation. A cut off there will give an approximate solution which is as good as an exact solution.
- Also, it was found out that in Muceli 2014 it is explained with a 192 channel grid that NMF is capable of explaining for the same sort of recreation if 16, 8 or 6 electrode configuration is used (muceli 2014). Therefore, the number of

electrodes being only 8 should not be a problem. Though, the doubt about data was still there that if the data is capable of reproducing the approximate solution.

- For that we generated a simulated data and used it to find if with 8 electrodes and 22 electrodes data nnmf is capable of giving an exact solution for any of these matrices. We found out that for 8 as well as 22 electrodes both the solution is not stable. Therefore 'nnmf' is not the way to go forward. Now we will check if a customized NMF algorithm will be able to produce an approximate solution for both 22 and 8 electrode data.

- **Simulated Data**
  - Simulated data was generated based on the law
  - $\bar{x} = g(\sum_{i=1}^{N} ci * \overline{Wi} + e)$: Type 1
  - Here, function g is a non-negative constraint. In Tresh (2006) it is imposed as making negative values zero, but we used it as taking the absolute value of the final matrix elements.
  - With 'nnmf' the solution achieved for N=8 and N=22 both were not stable.
  - With customized NMF the solution achieved is approximate and stable. Yet it fails sometimes because we are using pseudo random number generators to generate the data.
  - Note: The data generated is in the form of a Gaussian distribution. It is done so because histogram of original data shows that the original data is also following Gaussian distribution.
  - One surprising thing to notice with original data is that, when used original data instead of simulated data in customized NMF;

    data=EF_data.EF11.Trial_14.RAW;
    [W,H]=nmf_als(Xa,4,10000,1);
    [W1,H1]=nmf_als(Xa,4,12000,1);
    W for 10000 iteration and W1 for 12000 iteration produce an approximate result (columns are shuffled) . This is also near the global minimum as the sum of absolute value of '(X-Xr)' reaches less than 10^-11.
  - I still am looking forward to work on it further so that simulated data can also produce approximate solution
  - I changed the aforementioned simulated data law and produced a different type of simulated data, which can be given by;
  - $\bar{x} = g(\overline{Wi} * ci + e)$: Type 2

- Here, *ci* and $\overline{Wi}$ are generated at once instead of $\overline{Wi}$ generating in column wise manner. This gives a better representation of Gaussian data, Hence, similar to our original data.
- The results for Type 2 data are stable as of our original data. (shuffled columns)
- Hence, we can move forward with customized algorithm.

- **Working with a customized NMF algorithm**
  - Much already has been explained about customized NMF. It is much simpler in structure and contains a multiplicative update rule, which is also available in 'nnmf'. The ease with this algorithm is that max iteration alone can be controlled without giving any initial solution. Initial solution in 'nnmf' has to be provided by user. In customized NMF it is taken random and later gets updated with multiplicative update rule in next iteration.
  - Solution for W and H for all participants and trials is calculated. Yet I think that I should calculate it again for all and compare it with the previous result. This has not been done. It will give me surety if I should move forward with this result or not.
  - Further, NDP, CPA and cluster analysis has to be performed in upcoming days.

Results

- **Dimensionality analysis**
  - Based on max VAF using 'nnmf' it was decided that 4 muscle synergies are enough to reconstruct more than 77% of the original data.

mVAF =

Columns 1 through 9

0.8685	0.8504	0.8927	0.8778	0.9229	0.9289	0.9049	0.9031	0.8563
0.8899	0.8582	0.8945	0.8624	0.9029	0.8988	0.8802	0.9162	0.8300
0.9297	0.8580	0.9000	0.8622	0.8816	0.8869	0.8796	0.9233	0.8843
0.8353	0.8452	0.7885	0.8902	0.8800	0.8721	0.8224	0.8514	0.8800
0.8399	0.8362	0.8200	0.8785	0.8754	0.8872	0.7851	0.9022	0.8893
0.8873	0.8314	0.8141	0.9655	0.9030	0.8518	0.8385	0.8561	0.8788
0.8156	0.8471	0.8463	0.9272	0.9125	0.8328	0.8184	0.8741	0.8387
0.8428	0.8349	0.8478	0.9514	0.9182	0.8594	0.8730	0.8848	0.8756
0.8687	0.8375	0.8412	0.9389	0.9133	0.8723	0.8543	0.8574	0.8473

0.8143	0.8252	0.8290	0.9435	0.8737	0.8744	0.8536	0.8474	0.8341
0.9211	0.8179	0.8438	0.9331	0.9032	0.8728	0.8551	0.8289	0.8598
0.9097	0.8245	0.8278	0.9491	0.8827	0.8761	0.8371	0	0.8390
0.8240	0.8954	0.8307	0.9269	0.8954	0.8744	0.8581	0.8267	0.8316
0.8611	0.8984	0.8446	0.9263	0.8673	0.8744	0.8338	0.8283	0.8523
0.8888	0.8917	0.8506	0.8799	0.8684	0.8744	0.8189	0.8348	0.8588

Columns 10 through 18

0.9052	0.8675	0.8717	0.8759	0.8154	0.8606	0.8262	0.9508	0.8171
0.8640	0.8317	0.8924	0.8764	0.8376	0.8778	0.8289	0.9254	0.8461
0.8657	0.8829	0.9190	0.8767	0.8347	0.8456	0.8260	0.8842	0.8650
0.8350	0.8571	0.9487	0.8373	0.8467	0.8798	0.8129	0.8060	0.8635
0.8280	0.8840	0.9320	0.8542	0.8445	0.8878	0.8362	0.8923	0.8328
0.8703	0.8717	0.9259	0.8223	0.8421	0.8797	0.8482	0.8876	0.8357
0.8308	0.8420	0.9376	0.8533	0.8353	0.8268	0.8412	0.8551	0.8596
0.8491	0.8694	0.8981	0.8612	0.8466	0.8162	0.8564	0.8253	0.8376
0.8530	0.8790	0.8788	0.8234	0.8543	0.8368	0.8238	0.8496	0.8664
0.8146	0.8611	0.8785	0.8061	0.8482	0.8430	0.8337	0.7947	0.8429
0.8124	0.8326	0.8699	0.7914	0.8671	0.8520	0.8389	0.8163	0.8797
0.8087	0.8433	0.8523	0.7888	0.8401	0.8434	0.8225	0.8362	0.8628
0.8444	0.8962	0.8670	0.8278	0.8249	0.8399	0.8115	0.7980	0.8412
0.8565	0.8822	0.8655	0.8296	0.8507	0.8591	0.8010	0.8323	0.8933
0.8399	0.8562	0.8886	0.8999	0.8477	0.8502	0.8263	0.8318	0.8630

Columns 19 through 20

0.8658	0.8129
0.8762	0.8282
0.8552	0.8886
0.8752	0.8134
0.8897	0.8798
0.8822	0.8812
0.8342	0.8121
0.8490	0.7778
0.8756	0.7871
0.8612	0.8110
0.9032	0.8046
0.8970	0.7925

0.8577 0.8338
0.8358 0.7859
0.8322 0.8002

- With changing the NMF algorithm, we are also calculating the VAF of the approximate global solution again. This will make sure if the solution achieved is worthy of storing. So far more than 80% of recreation is achieved.

- **Weighting Matrix Comparison**
  To be updated after NDP comparisons.