

Linköping University Post Print

**A Brain Computer Interface for
Communication Using Real-Time fMRI**

Anders Eklund, Mats Andersson, Henrik Ohlsson, Anders Ynnerman and Hans Knutsson

N.B.: When citing this work, cite the original article.

©2010 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Anders Eklund, Mats Andersson, Henrik Ohlsson, Anders Ynnerman and Hans Knutsson, A Brain Computer Interface for Communication Using Real-Time fMRI, 2010, Proceedings from the 20th International Conference on Pattern Recognition (ICPR), 3665-3669.
<http://dx.doi.org/10.1109/ICPR.2010.894>

Postprint available at: Linköping University Electronic Press
<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-54038>

A Brain Computer Interface for Communication Using Real-Time fMRI

Anders Eklund*[§], Mats Andersson*[§], Henrik Ohlsson[†], Anders Ynnerman^{‡§} and Hans Knutsson*[§]

*Division of Medical Informatics, Department of Biomedical Engineering

[†]Division of Automatic Control, Department of Electrical Engineering

[‡]Division for Visual Information Technology and Applications, Department of Science and Technology

[§]Center for Medical Image Science and Visualization (CMIV)

{andek, matsa, knutte}@imt.liu.se, ohlsson@isy.liu.se, andyn@itn.liu.se

Linköping University, Sweden

Abstract—We present the first step towards a brain computer interface (BCI) for communication using real-time functional magnetic resonance imaging (fMRI). The subject in the MR scanner sees a virtual keyboard and steers a cursor to select different letters that can be combined to create words. The cursor is moved to the left by activating the left hand, to the right by activating the right hand, down by activating the left toes and up by activating the right toes. To select a letter, the subject simply rests for a number of seconds. We can thus communicate with the subject in the scanner by for example showing questions that the subject can answer. Similar BCI for communication have been made with electroencephalography (EEG). In these implementations the subject for example focuses on a letter while different rows and columns of the virtual keyboard are flashing. The system then tries to detect if the correct letter is flashing or not. In our setup we instead classify the brain activity. Our system is not limited to a communication interface, but can be used for any interface where five degrees of freedom is necessary.

Keywords—Brain computer interface, BCI, real-time fMRI, EEG, ANN, SVM

I. INTRODUCTION

In fMRI the objective is normally to estimate a level of brain activity in each voxel. The resulting activity map is for example used to study different functions of the brain or as an aid in surgical planning to prevent removal of important brain tissue. If the statistical analysis is performed in real-time, the result can be feed back to the subject in order to close the loop. The system is then often called a brain computer interface (BCI), since the brain and the computer are linked and can work together to solve a given task.

While previous fMRI based BCI have classified between two classes [1] or three classes [2], we instead classify between five different classes, to give the subject five degrees of freedom. We choose to use our system for communication, since EEG has been used for communication [3] and many have claimed that fMRI is far too slow for similar setups. fMRI however has the advantage of a higher spatial resolution than EEG and should therefore be better suited for BCIs that use a larger number of activities. Our system is not limited to communication but can be used for any interface where five degrees of freedom is necessary. Some work about communication using single-trial fMRI

has been made by Sorger et al. [4] where the subject could answer multiple choice questions by performing different activities. In their setup the subject is thus only able to give predetermined responses, while the subject in our setup can give any response. Our system might seem similar to the setup by Yoo et al. [5] where they perform spatial navigation in a labyrinth by imagination of different activities. The main advantage of our system is that we classify the brain activity every second, while Yoo et al. needed 2 minutes of data and 20 seconds of calculations for each step in the labyrinth. The difference between our setup and EEG based communication interfaces, is that we classify the brain activity, while the current EEG based methods rather detect a change in the EEG signal when the correct letter flashes, i.e. a classification if the correct letter flashes or not [6]. It is however of course possible to classify the brain activity from the EEG signal as well.

II. METHODS

We use a one layer artificial neural network (ANN) [7] with five output nodes to classify the brain activity. The ANN is first trained with a training dataset where the subject performs the five different activities. We use four periods of training, where each period consists of 20 seconds of left hand activity, 20 seconds of right hand activity, 20 seconds of left toes activity, 20 seconds of right toes activity and 20 seconds of rest. Hand and foot activity are easy to perform in the MR scanner and is therefore suitable for steering of the cursor. Since moving of the feet might introduce large head movements, we instead only move the toes. If we use four similar periods, this will result in biased training data since it will only contain 5 out of the 20 possible activity transitions. To get unbiased training data, we permute the sequence in each period such that the training data contains all the 20 transitions. The permuted training sequence can be written as 1, 2, 3, 4, 5, 1, 3, 5, 4, 2, 5, 3, 2, 1, 4, 3, 1, 5, 2, 4, 1 where the numbers represent the five different activities.

A. Artificial neural networks

In the training phase a volume of data was collected each second, resulting in 420 training vectors for the ANN.

A spatial mask was used to train the classifier on the brain voxels only, about 9000 of the 64 000 voxels were considered to be inside the brain. Since the dimension of the training vectors, 9000, is much higher than the number of training vectors, 420, using a non-linear activation function can lead to overfitting of the training data but this does not seem to be a problem in our case. We used a softmax activation function such that the output from the nodes of the ANN can be interpreted as posterior probabilities. The ANN is trained a number of epochs and in each epoch the ANN is given the current weights of the discriminant function and the correct classifications. The difference between the correct classifications and the calculated classifications is then used to adjust the weights by walking in the negative direction of the gradient with a certain step length n . This can be described with the following algorithm, where x is the data matrix containing all the training vectors, w are the weights for each output node and d is a matrix containing the information about the correct classes. We set d to 0.01 for the four wrong classes and 0.96 for the correct class.

```

for each epoch
   $y = w^T x$ 
  for each class  $c$ 
    for each timepoint  $t$ 
      
$$p(c, t) = \frac{\exp(y(c, t))}{\sum_{c=1}^5 \exp(y(c, t))}$$

    end timepoints
  end classes
   $w = w - n \cdot x \cdot (p - d)^T$ 
end epochs

```

We trained the ANN with the permuted sequence described above and used 500 epochs and a step length of $n = 0.00011$. We collect a number of datasets to train and evaluate the classifier. To see how the classifier handles new data, the validation is done in the real-time phase.

B. Motion compensation

Since the subject can move the head during the training and the real-time phase, it is important to apply motion compensation. Without the motion compensation it could be possible to learn to classify the different activities from the position of the head. There are a number of motion compensation algorithms, compared by Oakes et al. [8], that normally are used in fMRI. One problem with these algorithms is that since the voxel intensity itself is used, the increased intensity caused by the blood oxygen level dependent (BOLD) signal can disturb the registration algorithm. We instead use phase based registration, since the local phase, from for example quadrature filters, is invariant to a change of intensity. The concept of phase based registration is described by Hemmendorff et al. [9]. We use a more

simple version of phase based registration in order to be able to run it in real-time on a standard laptop. This version is described in our recent work [10] where we implemented the algorithm on the graphics processing unit (GPU), to enable fast phase based registration for large medical volumes.

C. Detrending

Since there are drifts and trends in the fMRI data, both from the MR scanner and from the subject, it is necessary to detrend the data prior to further analysis. We use a sliding window of 45 seconds and detrend each voxel time series separately by removing the mean of the voxel from the 45 latest volumes. Detrending is further explained by Friman et al. [11].

D. Experiment Setup

In the realtime setup the used MR scanner was a 1.5 T Philips Achieva. The acquisition resolution was 80 x 80 x 10 voxels where the voxels had a physical size of 3 x 3 x 3 mm. We used an echo time of 40 ms and a repetition time of 1 s. The motion compensation, detrending, classification of the brain activity and visualization of the virtual keyboard was done in Matlab on a laptop with an Intel 2.2 GHz Core 2 Duo processor and 4 GB of memory. The images of the subjects brain were collected through a network interface with the MR scanner. The motion compensation took about 0.3 s, the detrending 0.1 s and the classification 0.02 s, making the system fast enough to collect a volume of data each second.

The result from the classifier is a control signal for the cursor, if it should move to the left, to the right, up, down or remain still in order to select a letter. The current state of the communication interface was shown to the subject in a pair of VR goggles. In order to select a letter, the letter must first be highlighted and then the subject must rest for 4 seconds in a row.

III. RESULTS

We tested our system with a male healthy volunteer, subject 1, of age 49. The subject first conducted the training sequence and was then in the real-time phase given the possibility to answer a question. The user got the question 'Brand of your motorcycle?' and tried to write 'Guzzi' (which is short for the Italian motorcycle brand 'Moto Guzzi'). The subject could move the cursor in the four directions and select letters by resting. The communication interface and the resulting trajectory in the real-time phase is given in Figure 1. The user did not focus on to get the spelling right, since the current system is slow, but to prove that the setup works.

To further evaluate our classifier, we collected data from two additional male subjects (age 28 and 40). The used MR scanner for these subjects was a 3T Siemens Tim Trio. We used a repetition time of 2 s and an echo time of 30 ms.

Brand of your motorcycle?

GUZZZZII

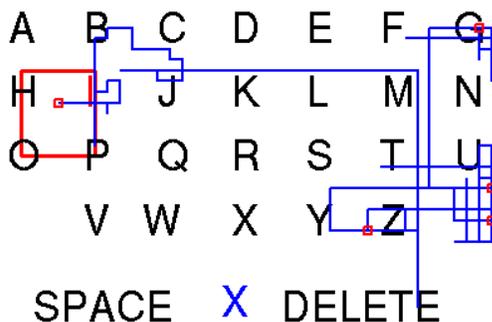


Figure 1. The communication interface that the subject sees in the VR goggles. The message from the person outside the MR scanner is given in green and the response from the subject is given in blue. The virtual keyboard consists of all the possible letters and functions for space and delete. The subject steers the cursor, the red square, and the letter that is currently active is highlighted in red. The letter between SPACE and DELETE indicates the current classification of the brain activity, D means down, U means up, L means left, R means right and X means rest. The blue line is the trajectory of the cursor in the real-time phase. The locations where the subject selected a letter is marked with a red square. The letter 'Z' was selected four times at the same location and 'I' two times at the same location. The total time for this trajectory is approximately 400 seconds.

The collected volumes have a resolution of $64 \times 64 \times 30$ voxels, where each voxel has the physical size of $3.75 \times 3.75 \times 4$ mm. For these volumes the number of brain voxels is about 25 000. We compared our ANN to the support vector machines (SVM) classifier that is implemented as 3D SVM, by LaConte et al. [1], in the AFNI software by Cox et al. [12]. Since SVM only can perform binary classification, 10 binary classifiers were trained (class 1 versus class 2, 1 versus 3, 1 versus 4, 1 versus 5, 2 versus 3, 2 versus 4, 2 versus 5, 3 versus 4, 3 versus 5, 4 versus 5). A directed acyclic graph was then used to make a classification of each fMRI volume. For the SVM classifier the preprocessing was done in AFNI while we used our own motion compensation and detrending for the ANN. We used a linear kernel for the SVM and therefore changed to a linear activation function for the ANN, in order to make an as equal comparison as possible.

We trained and evaluated the classifiers with two datasets from a non permuted training sequence (NP1, NP2) and two datasets from a permuted training sequence (P1, P2), for each of the three subjects. All datasets were 420 seconds long. The classification performance, as function of training dataset and evaluation dataset, for the ANN and SVM is given in Tables I, II and III. The resulting confusion matrices are given in Tables IV and V.

IV. DISCUSSION

We have taken the first step towards a BCI for communication using real-time fMRI. The subject in the MR scanner

controls a cursor on a virtual keyboard to select different letters that can form words. Our current system is rather slow and does not work flawlessly but proves that communication through real-time fMRI is feasible. One possible application for our setup is to help locked-in people communicate with the outside world. The advantage compared to EEG is that it would be possible to look at the brain activity at the same time as the communication [13], since fMRI has a much higher spatial resolution than EEG.

We have showed that our ANN classifier works well for three subjects, it achieves a mean classification rate of 88.3% for the 36 dataset combinations given in Tables I, II and III, while SVM achieves a mean classification rate of 75.9%. SVM performs slightly better than our ANN if we do binary classification of the 5 classes, but it can not handle the multiple class problem equally well since the 10 binary classifiers are trained separately. One possible reason for the lower classification rate with SVM is that the classes have to be independent for SVM to achieve optimal performance, but since for example bilateral activation is common, left and right hand and toe activation are not independent. We do not believe that the difference in classification performance is related to the different preprocessing of the fMRI data for ANN and SVM.

The permuted training sequence also seems to give a higher mean classification performance than the non permuted sequence. For the ANN the mean classification rate is 87.1% when a non permuted training sequence is used for training and 89.4% when a permuted training sequence is used. For SVM the corresponding mean classification rates are 74.2% and 77.7%.

By looking at the confusion matrices in Tables IV and V, we see that the big difference between the ANN and SVM is the classification of the toe activities and the rest state. One possible explanation for this is that the toe activity is much closer to the corpus callosum, compared to the hand activity, where the left and the right hemispheres meet and thereby the activity overlap for left and right activity might be bigger. The ANN handles the classification of rest better since it looks at the 5 classes at the same time.

While the navigation of the cursor works as expected, the hard thing is to stop the cursor at the right letter and select it. The classifier is very stable when the same activity is applied for a longer time, this can be seen in Figure 1 by looking at the trajectory from 'Z' to 'I'. It is however the shorter times of activity that are harder to classify correctly. It might not be optimal to use a row-column layout for the virtual keyboard. Friman et al. [14] showed that a rhombus layout was a better than a row-column layout for an EEG based BCI for spelling.

Another problem with the setup is that each command from the subject is delayed a number of seconds, due to the delay of the BOLD signal. The BOLD delay is hard to remove since it is a part of the human physiology. One

Table I
CLASSIFICATION PERFORMANCE FOR ANN AND SVM AS FUNCTION OF TRAINING DATASET AND EVALUATION DATASET FOR SUBJECT 1.

| Training data / Evaluation data | NP1 / NP2 | NP2 / NP1 | NP1 / P1 | NP1 / P2 | NP2 / P1 | NP2 / P2 |
|---------------------------------|-----------|-----------|----------|----------|----------|----------|
| ANN | 93.1% | 94.8% | 89.2% | 81.6% | 87.6% | 85.9% |
| SVM | 83.3% | 82.8% | 78.2% | 65.6% | 81.1% | 71.8% |
| | | | | | | |
| Training data / Evaluation data | P1 / P2 | P2 / P1 | P1 / NP1 | P2 / NP1 | P1 / NP2 | P2 / NP2 |
| ANN | 93.1% | 93.5% | 90.2% | 90.0% | 90.0% | 91.9% |
| SVM | 85.2% | 83.3% | 78.7% | 73.9% | 74.6% | 70.1% |

Table II
CLASSIFICATION PERFORMANCE FOR ANN AND SVM AS FUNCTION OF TRAINING DATASET AND EVALUATION DATASET FOR SUBJECT 2.

| Training data / Evaluation data | NP1 / NP2 | NP2 / NP1 | NP1 / P1 | NP1 / P2 | NP2 / P1 | NP2 / P2 |
|---------------------------------|-----------|-----------|----------|----------|----------|----------|
| ANN | 88.1% | 91.9% | 87.1% | 88.1% | 88.1% | 86.7% |
| SVM | 78.6% | 86.2% | 70.9% | 68.1% | 78.6% | 77.1% |
| | | | | | | |
| Training data / Evaluation data | P1 / P2 | P2 / P1 | P1 / NP1 | P2 / NP1 | P1 / NP2 | P2 / NP2 |
| ANN | 91.9% | 91.0% | 90.5% | 89.1% | 88.1 % | 85.2% |
| SVM | 87.1% | 83.8% | 81.4% | 75.2% | 82.4% | 76.7% |

Table III
CLASSIFICATION PERFORMANCE FOR ANN AND SVM AS FUNCTION OF TRAINING DATASET AND EVALUATION DATASET FOR SUBJECT 3.

| Training data / Evaluation data | NP1 / NP2 | NP2 / NP1 | NP1 / P1 | NP1 / P2 | NP2 / P1 | NP2 / P2 |
|---------------------------------|-----------|-----------|----------|----------|----------|----------|
| ANN | 85.7% | 86.2% | 83.8% | 83.8% | 82.9% | 83.3% |
| SVM | 70.0% | 66.2% | 67.6% | 71.0% | 67.6% | 70.0% |
| | | | | | | |
| Training data / Evaluation data | P1 / P2 | P2 / P1 | P1 / NP1 | P2 / NP1 | P1 / NP2 | P2 / NP2 |
| ANN | 90.0% | 90.5% | 88.6% | 85.7% | 85.7% | 84.3% |
| SVM | 76.2% | 79.1% | 72.9% | 73.8% | 72.9% | 71.0% |

Table IV
CONFUSION MATRIX FOR ALL SUBJECTS AND ALL DATASETS FOR ANN

| ANN | Actual left hand | Actual right hand | Actual left toes | Actual right toes | Actual rest |
|----------------------|------------------|-------------------|------------------|-------------------|-------------|
| Predicted left hand | 92.8% | 2.9% | 1.5% | 1.9% | 8.0% |
| Predicted right hand | 2.0% | 91.7% | 3.0% | 0.7% | 3.7% |
| Predicted left toes | 0.2% | 2.3% | 90.2% | 7.3% | 2.7% |
| Predicted right toes | 1.0% | 2.0% | 4.0% | 87.1% | 6.7% |
| Predicted rest | 4.0% | 1.1% | 1.3% | 3.0% | 78.9% |

Table V
CONFUSION MATRIX FOR ALL SUBJECTS AND ALL DATASETS FOR SVM

| SVM | Actual left hand | Actual right hand | Actual left toes | Actual right toes | Actual rest |
|----------------------|------------------|-------------------|------------------|-------------------|-------------|
| Predicted left hand | 85.4% | 3.9% | 2.1% | 2.7% | 11.9% |
| Predicted right hand | 4.6% | 87.5% | 9.6% | 4.1% | 8.6% |
| Predicted left toes | 1.1% | 3.2% | 72.5% | 19.5% | 6.6% |
| Predicted right toes | 3.2% | 2.5% | 13.8% | 68.8% | 8.9% |
| Predicted rest | 5.7% | 2.9% | 2.0% | 4.9% | 64.0% |

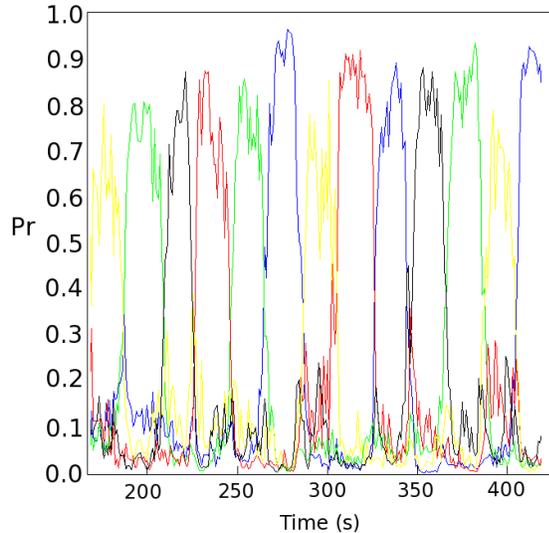


Figure 2. The output from the nodes in the ANN for a number of timepoints for one of the permuted datasets for subject 1. The five different activities are represented with different colours. It is clear that the probability of the current activity, from the softmax activation function, decreases at the same time as the probability of the new activity increases and this can be used in order to faster detect a change of the brain activity, instead of using the current maximum probability only.

possible way to faster detect a change of brain activity is to look at the output from the nodes in the ANN a number of seconds backwards, instead of only using the current maximum probability. As can be seen in Figure 2, the probability of the current activity drops at the same time as the the probability of the new activity rises. By thus detecting a negative and a positive derivative at the same time, it would possible to detect a change of brain activity several seconds earlier than by just looking at the current maximum probability. A more sophisticated approach might be to use recurrent neural networks [15], such that a state can be given at each timepoint, and not only a classification.

ACKNOWLEDGMENT

We thank Stephen Laconte et al. at the Department of Neuroscience, Baylor college of medicine (Houston, Texas, USA), for letting us use one of their 3T MR scanners for data collection and helping us with their 3D SVM. This work was supported by the Strategic Research Centers MOVIII and CADICS, funded by the Swedish Foundation for Strategic Research, SSF.

REFERENCES

[1] S. Laconte, S. Peltier, and X. Hu, "Real-time fMRI using brain-state classification," *Human Brain Mapping*, vol. 28, pp. 1033–1044, 2007.

[2] A. Eklund, H. Ohlsson, M. Andersson, J. Rydell, A. Ynnerman, and H. Knutsson, "Using Real-Time fMRI to Control a Dynamical System by Brain Activity Classification," *Lecture*

Notes in Computer Science, Proceedings of Medical Image Computing and Computer Assisted Intervention (MICCAI), vol. 5761, pp. 1000–1008, 2009.

- [3] J. Wolpaw, N. Birbaumer, D. McFarland, G. Pfurtscheller, and T. Vaughan, "Brain-computer interfaces for communication and control," *Clinical Neurophysiology*, vol. 113, pp. 767–791, 2002.
- [4] B. Sorger, B. Dahmen, J. Reithler, O. Gosseries, A. Maudoux, S. Laureys, and R. Goebel, "Another kind of 'BOLD response': answering multiple-choice questions via online decoded single trial brain signals," *Progress in Brain Research*, vol. 177, pp. 275–292, 2009.
- [5] S. Yoo, T. Fairney, N. Chen, S. Choo, L. Panych, H. Park, S. Lee, and F. Jolesz, "Brain-computer interface using fMRI: spatial navigation by thoughts," *Neuroreport*, vol. 15, pp. 1591–1595, 2004.
- [6] T. Manoj, C. Guan, and J. Wu, "Robust classification of EEG signal for brain-computer interface," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 14, pp. 24–29, 2006.
- [7] S. Haykin, *Neural networks - a comprehensive foundation*. Prentice Hall, 2009, ISBN 0-13-273350-1.
- [8] T. Oakes, T. Johnstone, K. Walsh, L. Greischar, A. Alexander, A. Fox, and R. Davidson, "Comparison of fMRI motion correction software tools," *Neuroimage*, vol. 28, pp. 529–543, 2005.
- [9] M. Hemmendorff, M. Andersson, T. Kronander, and H. Knutsson, "Phase-based multidimensional volume registration," *IEEE Transactions on Medical Imaging*, vol. 21, pp. 1536–1543, 2002.
- [10] A. Eklund, M. Andersson, and H. Knutsson, "Phase based volume registration using CUDA," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2010.
- [11] O. Friman, M. Borga, P. Lundberg, and H. Knutsson, "Detection and detrending in fMRI data analysis," *Neuroimage*, vol. 22, pp. 645–655, 2004.
- [12] R. Cox, "AFNI: Software for analysis and visualization of functional magnetic resonance neuroimages," *Computers and Biomedical Research*, vol. 29, pp. 162–173, 1996.
- [13] M. Boly, M. Coleman, M. Davis, A. Hampshire, D. Bor, G. Moonen, P. Maquet, J. Pickard, S. Laureys, and A. Owen, "When thoughts become action: An fMRI paradigm to study volitional brain activity in non-communicative brain injured patients," *Neuroimage*, vol. 36, pp. 979–992, 2007.
- [14] O. Friman, T. Luth, I. Volosyak, and A. Gräser, "Spelling with steady-state visual evoked potentials," in *Proceedings of the 3rd International IEEE EMBS Conference on Neural Engineering*, 2007.
- [15] F. J. Pineda, "Generalization of back-propagation to recurrent neural networks," *Phys. Rev. Lett.*, vol. 59, no. 19, pp. 2229–2232, 1987.