Prediction of Movements by Online Analysis of Electroencephalogram with Dataflow Accelerators

Hendrik Woehrle[1], Johannes Teiwes[2], Marc Tabie[1,2], Anett Seeland[1], Elsa Andrea Kirchner[1,2] and Frank Kirchner[1,2]

[1]*German Research Center for Artificial Intelligence (DFKI), Robotics Innovation Center, Bremen, Germany*

[2]*AG Robotik, University of Bremen, Bremen, Germany*

Keywords: Brain Computer Interfaces, Mobile Systems, FPGA, Hardware Accelerator.

Abstract: Brain Computer Interfaces (BCIs) allow to use psychophysiological data for a large range of innovative applications. One interesting application for rehabilitation robotics is to modulate exoskeleton controls by predicting movements of a human user *before* they are actually performed. However, usually BCIs are used mainly in artificial and stationary experimental setups. Reasons for this are, among others, the immobility of the utilized hardware for data acquisition, but also the size of the computing devices that are required for the analysis of the human electroencephalogram. Therefore, mobile processing devices need to be developed. A problem is often the limited processing power of these devices, especially if there are firm time constraints as in the case of movement *prediction*. Field programmable gate array (FPGA)-based application-specific dataflow accelerators are a possible solution here. In this paper we present the first FPGA-based processing system that is able to predict *upcoming* movements by analyzing the human electroencephalogram. We evaluate the system regarding computation time and classification performance and show that it can compete with a standard desktop computer.

1 INTRODUCTION

The prediction of human movements by online analysis of the electroencephalogram (EEG) is a frequent task in Brain Computer Interfaces (BCIs). The prediction of movements can be used in various applications, such as assistive devices like orthoses and rehabilitation robotics (Pfurtscheller, 2000; Ahmadian et al., 2013; Kirchner et al., 2013a; Kirchner et al., 2013c) or in telemanipulation devices (Folgheraiter et al., 2011; Folgheraiter et al., 2012; Seeland et al., 2013; Lew et al., 2012).

Different event related patterns can be found in the EEG before a movement is actually performed. These are usually related to neuronal processes related to movement preparation e.g. specific frequency components in the EEG reflecting neural synchronization or desynchronization (ERD/ERS) (Bai et al., 2011) or movement related cortical potentials (MRCPs) such as the lateralized readiness potential (LRP) (Blankertz et al., 2006).

However, for a *reliable* detection of upcoming movements a range of complex signal processing methods have to be applied to detect the relevant potentials in the raw data. Obviously, all these operations have to be performed online and in real-time, i.e. the *movement predictions* have to be available *before* the *real movements* are executed in order to be useful in applications.

1.1 Mobile and Miniaturized Brain Computer Interfaces

Many applications require that the signal processing is performed in small devices that are *embedded* into the specific environment. For different medical purposes or rehabilitation approaches the *disappearance* of computing devices by means of an integration of these computers into anyway present medical devices or other equipment would be beneficial.

In order to integrate the computing devices into other systems, they need to have small physical dimensions and a low power consumption (in order to be able to use small accumulators). However, since the employed signal processing operations for the detection of specific patterns or potentials in EEG data are often computationally expensive, current mobile

Wöhrle H., Teiwes J., Tabie M., Seeland A., Kirchner E. and Kirchner F..
Prediction of Movements by Online Analysis of Electroencephalogram with Dataflow Accelerators.
DOI: 10.5220/0005139400310037
In *Proceedings of the 2nd International Congress on Neurotechnology, Electronics and Informatics* (NEUROTECHNIX-2014), pages 31-37
ISBN: 978-989-758-056-7

BCI systems rely only on a small number of electrodes (Wang et al., 2013; Webb et al., 2012) which are sufficient for *simple* approaches that are based on, e.g., the detection of steady-state visually evoked potentials (Wang et al., 2013; Chi et al., 2012). Even field programmable gate array (FPGA)-based systems have these shortcomings (Khurana et al., 2012; Shyu et al., 2013).

These approaches are not sufficient if patterns must be detected that require the usage of several electrodes as it likely is the case in complex rehabilitation or telemanipulation applications (Kirchner et al., 2013b). Accordingly, specialized signal-processing systems that use complex algorithms and apply these in an online-fashion and real-time are needed here.

1.2 Overview about the Paper

In this paper we show the first system that fulfills the mentioned requirements by using application-specific dataflow accelerators which are realized as hardware components in programmable logic. In Section 2, the general hard- and software architecture is described. Subsequently, in Section 3 we present the experimental procedures that were employed to acquire data that we used for the evaluation of the system. The obtained results of the evaluation are then discussed in Section 4. The conclusion and future directions are finally given in Section 5.

2 HARD- AND SOFTWARE ARCHITECTURE

The application that is considered here results in two types of requirements. On the one hand side, the system for movement prediction has to be included into a complex environment, communicate with different other systems and provide various features for users like e.g. data recording and provision of configuration options. The functionalities to fulfill these requirements are very diverse, but usually not time-critical nor computationally expensive. Accordingly, we implement these in software (SW). On the other hand, a fixed set of signal processing algorithms has to be applied in a very short time frame for the data-analysis. Hence, we implement this part of the system as application-specific dataflow accelerators (DFAs), which can be realized as hardware components by programmable logic.

2.1 Hardware and Electronics Architecture

We developed our own electronics device *ZynqBrain* that we used as the central platform for processing in our experiments (Figure 1). The printed circuit board that is used for the *ZynqBrain* is manufactured in Pico-ITX format (7cm×10cm size). The main component is a Xilinx Zynq 7030 processing platform that consists of a Dual-Core ARM Cortex-A9 processor (operating at 666 MHz) and a programmable logic (PL) section (operating at 100 MHz in our setup), and is therefore well-suited for the required SW/HW partitioning of our system. The device contains an USB interface in order to be able to connect it directly to typical EEG-acquisition hardware, a SDHC card interface to store the software and EEG data, as well as a Gigabit Ethernet interface in order to transfer either data or results to other systems. Furthermore, the device contains low voltage differential signaling (LVDS) based interfaces to extend it in future with complemental electronics boards. The CPU runs a customized linux kernel (version 3.12) with a Linaro (Linaro, 2013) user space.

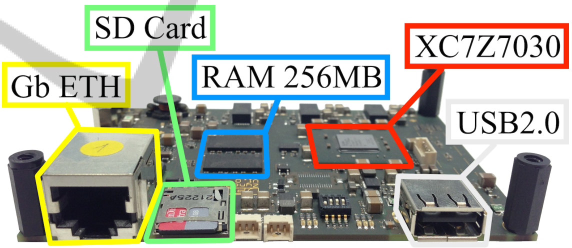

Figure 1: The *ZynqBrain* electronics board (pico-ITX form factor, 100 × 72 mm) with highlighted central components.

2.2 Software Architecture

Since the device is able to run usual software on the CPU part, it is able to use the signal processing software framework pySPACE (Krell et al., 2013) as the high-level processing and infrastructure software in our system. It contains modules for signal processing and machine learning as well as modules for service functionalities, e.g. configuration of the data acquisition or read previously stored data and to perform evaluations and store the results. Numpy and Scipy (Jones et al., 2001) are used by pySPACE as libraries, for e.g. matrix algebra and filtering (in case they were performed in SW, e.g. for comparison).

2.3 Dataflow Hardware Accelerator

The PL-section of the Z7030 can be utilized as an FPGA, i.e. it can be configured so that parts of it work as a specialized hardware component. Since these are specialized on performing a specific task, they are usually more efficient than a corresponding software implementation. We defined DFAs that implement exactly the signal processing and machine learning algorithms that are required for the online movement prediction. While the generic software tasks are executed on the CPU, the actual execution of the signal processing tasks are delegated to the PL part. The signal processing inside the FPGA uses a static dataflow principle, i.e. the hardware accelerator is implemented by a set of fixed circuits and the data is transformed while *flowing* through them. Figure 2 shows the data flow between software and hardware partitions. The DFAs are connected to the *host* CPU via AXI-Lite busses. Different parameters can be configured using a set of software-accessible registers. In order to process the EEG data, it is copied into the input FIFO buffers and the results are collected from the output FIFO buffers or result register. This setup has the advantage that the host CPU is *not* involved in the computations which are performed in the DFAs, and is therefore not occupied by these.

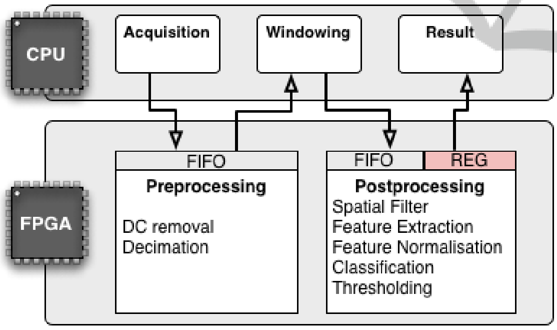

Figure 2: Dataflow among the CPU and FPGA part in the mobile setup with DFAs.

3 EXPERIMENTAL EVALUATION

In order to evaluate our system regarding prediction accuracy and processing speed, we assembled an experimental setup that allows us to compare *true* movement onsets to the predictions of the EEG-based system. Therefore, we evaluated our system with data that was acquired in a setup described in the following.

3.1 Experimental Setup

Eight healthy male subjects (age 29.9 ± 3.3 years) with normal or corrected to normal vision participated in this study. All participants gave written consent for participation and the study was approved by the

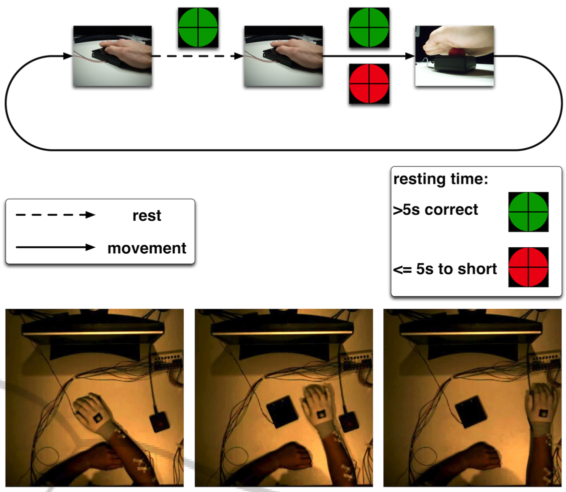

Figure 3: Illustration of the conducted experiments. Top: visualization of the paradigm. Bottom: three pictures of the setup, showing from left to right: the resting phase, the movement phase and the end of the movement.

ethics committee from the University of Bremen. Figure 3 shows the experimental setup and procedures. The subjects were seated in a chair in front of a table. A monitor, a flat board and a buzzer were placed on the table. The distance between buzzer and board was approximately 30 cm. The study consisted of one recording session per subject, each recording session was divided into three runs. The subjects were asked to perform 40 self-paced, voluntary movements of their right arm from the flat board to the buzzer and back. For the experimental implementation the software Presentation (Neurobehavioral Systems, Inc.) was used. During the experiments a green circle with a black fixation cross was shown to the subjects for minimizing the occurrence of eye-artifacts. The only restriction within the experiment was a fixed resting time of at least 5 s in between two consecutive movements. Movements that were performed too early were reported to the subjects and not taken into account for data analysis. A run was finished after 40 correct movements. For each subject, three runs were recorded in each session, resulting in 120 movements.

3.2 Data Acquisition

EEG was recorded at 5000 Hz with a 128 electrode (extended $10-20$ system) actiCAP system using four 32 channel BrainAmp DC amplifiers (BrainProducts GmbH, Munich, Germany). During recording the signals were filtered between 0.1 and 1000 Hz. Electrodes I1, OI1h, OI2h, and I2 were used for recording the electrooculogram, which was not considered in the following analysis. A motion tracking system was used to track a marker placed on the back of the subjects right hand. The system consisted of three ProReflex 1000 cameras (Qualisys AB, Gothenburg, Sweden). Motions of the hand were sampled at 500 Hz and a trigger signal was used to synchronize tracking data and EEG data. The movement onsets were extracted from the tracking data in an offline analysis. These movements onsets, however, were only used to infer the *true* movement onsets that can serve as a reference standard for the comparison with EEG-based predictions in the subsequent evaluations.

3.3 EEG Processing

All described analyses were performed offline and subject-wise, i.e., a 3-fold cross-validation scheme was used, where in each split two of the three runs were used for training of the machine learning and data-dependent signal processing methods, and the remaining run for testing. No testing data was used in the training phase and vice versa. 64 (extended $10-20$ system) of the 128 EEG channels were used in the analysis due to performance constraints that account only to the acquisition of the signals via USB. All data was processed in SW and in HW in a similar manner, i.e. the same algorithms were used and the time consumption and quality of the results (by comparing the classification accuracy). However, the SW implementations used double-precision floating-point arithmetic, while the HW counterparts are based on fixed-point arithmetic. The data was preprocessed in three parts, each consisting of several processing steps:

3.3.1 Preprocessing

First, the slowly-varying direct current offset was removed by a infinite impulse response filter. Next, the sampling rate was decimated from 5 kHz to 20 Hz in two steps (with an intermediate sampling rate of 125 Hz). The anti-alias finite impulse response filter of the second step was parameterized so that all frequencies greater than 4.0 Hz were attenuated. The Xilinx FIR Compiler was used for the finite impulse response (FIR) filter realization in the HW partition.

3.3.2 Windowing

Before the data can be processed by a spatial filter or classifier, it must be divided into distinct *instances*, that are processed independently from each other. Therefore, windows of the same length, i.e., 200 ms of duration were cut out of the data stream. Predictions were performed every 50 ms, so adjacent windows overlapped by 150 ms. For training of the spatial filter, classifier and feature standardization (see below), the windows of the training phase were labeled as related to a *movement* or to a *no movement* phase, based on the movement onsets found in the motion tracking data. Windows extracted from the interval between -4 s to -1 s were assumed to belong to the *no movement* class, and windows from -0.95 s to 0 s to the *movement* class. The Passive Aggressive Perceptron variant 1 (PA-1) (Crammer et al., 2006) was used for classification. For training of the classifier only the windows $[0,-0.2]$ s and $[-0.05,-0.25]$ s for the *movement* class and all of the *no movement* class were used, since we assume that these windows contribute most to the LRP, which the classifier shall detect (Kirchner et al., 2013a; Kirchner et al., 2013b). This is especially true for the *movement* windows since the LRP has its peak right at the beginning of a physical movement. Therefore only the two above mentioned windows close to the physical movement onset were used as training instances for the *movement* class. For evaluation purposes, the movement has to be detected within -1 s to -0.05 s. Before that, windows account for the *no movement* class and after that for the *movement* class.

3.3.3 Feature Extraction and Classification

For further data reduction the xDAWN spatial filter (Rivet et al., 2009) was applied to decrease the number of remaining channels to four, which can be realized as a matrix multiplication (using DSP48 (Xilinx Corporation, 2014) slices if realized in HW). Data from the remaining channels were merged to one feature vector and standardized. We used the PA-1 for classification (which results in the computation of a dot product - using DSP48 slices in the HW realization).

3.4 Evaluation Procedures

As stated above, we used pySPACE (Krell et al., 2013) for processing and evaluation. We compared our system with a standard PC that contains an 8-core Intel(R) Core(TM) i7 CPU 950 that was running at 3.07GHz and a Linux Mint operating system. We used four different computing setups in our comparison:

1. A single core of the standard desktop PC. In this case we used only a single CPU core for the processing.
2. A multi-core standard desktop PC. In this case we used the same system as before, but we used all 8 cores in order to apply the anti-alias filter in parallel channel-wise. This parallelization was performed using OpenMP (OpenMP, 2014).

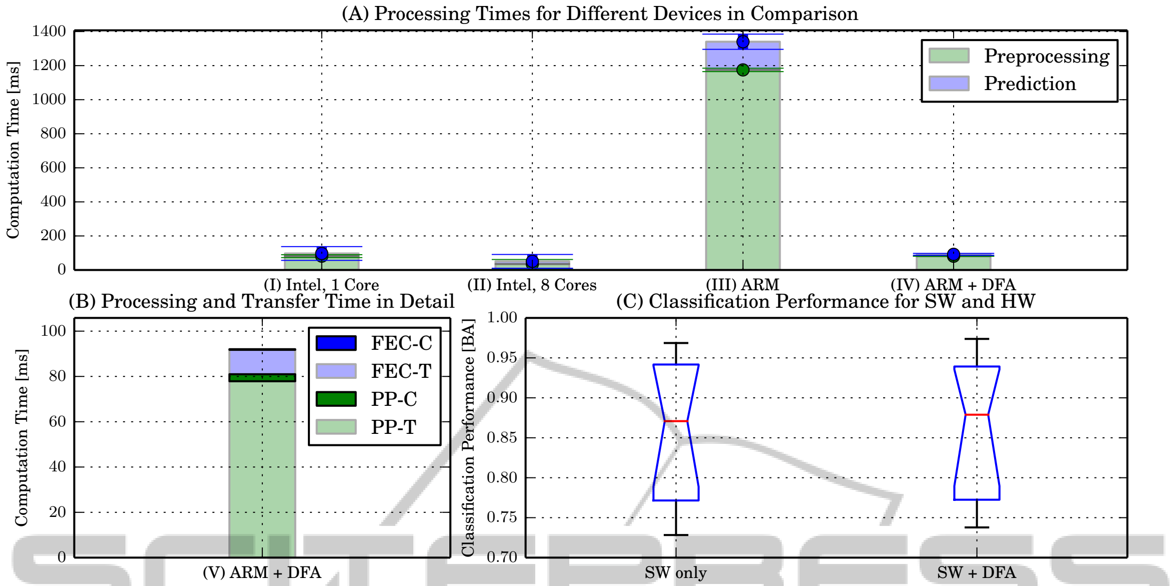

Figure 4: Classification and computation time performance for different computing setups for the analysis of 1 s of EEG data. (A): processing time for different computing devices; (B:) more detailed description of the computation/data transfer times if the DFAs are used (with PP-T: time for the data transfer to and from the DFA of the preprocessing part, PP-C: for the actual computations, FEC-T and FEC-C: corresponding times for the feature extraction and classification part; (C) classification accuracy for software only and software for high level tasks and signal processing tasks performed in the DFAs.

3. A single core of a mobile CPU. In this case we used only the ARM CPU part of the Zynq SoC.
4. The mobile CPU combined with application specific dataflow accelerators that are realized as hardware components in the programmable logic partition of the *Zynq*. The basic setup was as before, but the data processing was performed using the DFAs.

We used different schemes for SW/HW mapping, depending on the evaluation procedure:

- Cases 1 to 3: without DFA. In this case the high level data preprocessing, feature extraction and classification in the training as well as in the application phase was completely executed in SW.
- Case 4: with DFA. The preprocessing as well as the feature extraction and the classification in the test phase were performed using the DFA. However, the computation of the data-dependent parameters in the training phase were computed in SW.

4 RESULTS AND DISCUSSION

The results regarding the balanced classification accuracy (BA) and processing times are shown in Figure 4. The times correspond to the time that is required to process 1 s of EEG data in order to relate the different parts of the processing (preprocessing and to perform 20 predictions, since one prediction is performed every 50 ms).

It can be observed that the processing times of the desktop PC are fast enough for online prediction (A), i.e. they require less than 1 s of time to process 1 s of EEG data, which is not the case for the mobile processor. However, if the DFAs are used, the computation times can be dramatically reduced and the time constraints for real-time prediction are met. As shown in (B), most time is consumed for the data *transfer*, and not for the computations themselves. We expect that this can be accelerated in the future by using direct memory access for the data transfer in order to further decrease the latency.

Since fixed-point computations are used inside the DFAs, a major concern was that this might compromise the prediction accuracy. As shown in (C), this is not the case - the classification accuracy is not affected by the fixed point computations.

The performed evaluations were performed in a quite artificial setup that was designed in order to eliminate any kind of disturbances that could cause artifacts in the EEG data and to be able to reliably determine movement onsets with different methods as a gold reference for the evaluation process. However, the methodology for the detection of the MRCPs was already successfully applied in other, more challenging real-world application, i.e. the usage of MRCP-based movement prediction to enhance the user-experience of an exoskeleton by adapting the control algorithms (Seeland et al., 2013). Therefore, a required next step is the integration and application of the developed device in such an environment.

5 CONCLUSIONS AND FUTURE WORK

We showed that it is possible to use FPGA-based application-specific DFAs for the online analysis of the EEG in order to predict movements of humans *before* they are executed. We showed that the fixed-point arithmetic of the DFAs does not compromise the classification accuracy, but instead results in a high speedup of the processing time (in comparison with the mobile CPU without DFAs). This will allow us to integrate our systems into complex applications like robotic rehabilitation scenarios (Kirchner et al., 2013a).

In the future, we want to 1) enhance our system further by accelerating the data transfer to the DFAs by using direct memory access, 2) extend our system to multi-modal data processing, i.e., integrate the analysis of other physiological signals like the EMG (Kirchner et al., 2013c) into the system and allow the detection of other potentials, such as the P300 event related potential, 3) achieve user independence by integrating adaptive methods, and 4) use the device in more challenging real-world applications, e.g., integrate the *Zynqbrain* into an exoskeleton to perform *embedded* movement prediction.

ACKNOWLEDGEMENTS

Work was funded by the German Ministry of Economics and Technology (grant no. 50 RA 1011 and grant no. 50 RA 1012).

REFERENCES

Ahmadian, P., Cagnoni, S., and Ascari, L. (2013). How capable is non-invasive EEG data of predicting the next movement? A mini review. *Frontiers in Human Neuroscience*.

Bai, O., Rathi, V., Lin, P., Huang, D., Battapady, H., Fei, D.-Y., Schneider, L., Houdayer, E., Chen, X., and Hallett, M. (2011). Prediction of human voluntary movement before it occurs. *Clinical Neurophysiology*.

Blankertz, B., Dornhege, G., Lemm, S., Krauledat, M., Curio, G., and Müller, K.-R. (2006). The Berlin Brain-Computer Interface: machine learning based detection of user specific brain states. *Journal of Universal Computer Science*.

Chi, Y. M., Wang, Y.-T., Wang, Y., Maier, C., Jung, T.-P., and Cauwenberghs, G. (2012). Dry and noncontact EEG sensors for mobile brain-computer interfaces. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*.

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., and Singer, Y. (2006). Online passive-aggressive algorithms. *J. Mach. Learn. Res.*

Folgheraiter, M., Jordan, M., Straube, S., Seeland, A., Kim, S. K., and Kirchner, E. A. (2012). Measuring the improvement of the interaction comfort of a wearable exoskeleton. *International Journal of Social Robotics*.

Folgheraiter, M., Kirchner, E. A., Seeland, A., Kim, S. K., M., J., Woehrle, H., Bongardt, B., Schmidt, S., Albiez, J., and Kirchner, F. (2011). A multimodal brain-arm interface for operation of complex robotic systems and upper limb motor recovery. In *Proceedings of the 4th International Conference on Biomedical Electronics and Devices (BIODEVICES-11)*.

Jones, E., Oliphant, T., and Peterson, P. (2001). SciPy: Open source scientific tools for Python.

Khurana, K., Gupta, P., Panicker, R. C., and Kumar, A. (2012). Development of an FPGA-based real-time P300 speller. In *22nd International Conference on Field Programmable Logic and Applications (FPL)*.

Kirchner, E. A., Albiez, J., Seeland, A., Jordan, M., and Kirchner, F. (2013a). Towards assistive robotics for home rehabilitation. In *Proceedings of the 6th International Conference on Biomedical Electronics and Devices (BIODEVICES-13)*, Barcelona.

Kirchner, E. A., Kim, S. K., Straube, S., Seeland, A., Woehrle, H., Krell, M. M., Tabie, M., and Fahle, M. (2013b). On the applicability of brain reading for predictive human-machine interfaces in robotics. *PLoS ONE*.

Kirchner, E. A., Tabie, M., and Seeland, A. (2013c). Multimodal movement prediction - towards an individual assistance of patients. *PLoS ONE*.

Krell, M. M., Straube, S., Seeland, A., Woehrle, H., Teiwes, J., Metzen, J. H., Kirchner, E. A., and Kirchner, F. (2013). pySPACE - A Signal Processing and Classification Environment in Python. *Frontiers in Neuroinformatics*.

Lew, E., Chavarriaga, R., Silvoni, S., and Millán, J. D. R. (2012). Detection of self-paced reaching movement intention from EEG signals. *Frontiers in Neuroengineering*, 5:13.

Linaro (2013). Open source software for ARM SoCs. Technical report.

OpenMP (2014).

Pfurtscheller, G. (2000). Brain oscillations control hand orthosis in a tetraplegic. *Neuroscience Letters*, 292(3):211–214.

Rivet, B., Souloumiac, A., Attina, V., and Gibert, G. (2009). xDAWN algorithm to enhance evoked potentials: Application to brain computer interface. *Biomedical Engineering, IEEE Transactions on*.

Seeland, A., Woehrle, H., Straube, S., and Kirchner, E. A. (2013). Online movement prediction in a robotic application scenario. In *6th International IEEE/EMBS Conference on Neural Engineering (NER)*.

Shyu, K.-K., Chiu, Y.-J., Lee, P.-L., Lee, M.-H., Sie, J.-J., Wu, C.-H., Wu, Y.-T., and Tung, P.-C. (2013). Total design of an FPGA-based brain-computer interface control hospital bed nursing system. *Industrial Electronics, IEEE Transactions on*.

Wang, Y.-T., Wang, Y., Cheng, C.-K., and Jung, T.-P. (2013). Developing stimulus presentation on mobile devices for a truly portable SSVEP-based BCI. In *Engineering in Medicine and Biology Society (EMBC), 35th Annual International Conference of the IEEE*.

Webb, J., Xiao, Z.-G., Aschenbrenner, K. P., Herrnstadt, G., and Menon, C. (2012). Towards a portable assistive arm exoskeleton for stroke patient rehabilitation controlled through a brain computer interface. In *Biomedical Robotics and Biomechatronics (BioRob), 4th IEEE RAS EMBS International Conference on*, pages 1299–1304.

Xilinx Corporation (2014). UG479: 7 Series DSP48E1 Slice User Guide.