

Rapid Prototyping of an EEG-based Brain-Computer Interface (BCI)

Christoph Guger¹, Alois Schlögl¹, Christa Neuper², Dirk Walterspacher¹, Thomas Strein¹,
Gert Pfurtscheller^{1,2}

¹Institute for Biomedical Engineering, Department of Medical Informatics

²Ludwig-Boltzmann Institute for Medical Informatics and Neuroinformatics

University of Technology Graz

Inffeldgasse 16a, 8010 Graz, Austria

Tel.: (+43) 316 8735301

Fax: (+43) 316 8735349

e-mail: guger@dpmi.tu-graz.ac.at

pfu@dpmi.tu-graz.ac.at

Index Terms - brain-computer interface (BCI), event-related desynchronization (ERD), rapid prototyping, real-time software, single-trial EEG classification

This work was supported by the "Fonds zur Förderung der wissenschaftlichen Forschung", project P11208-MED, by the "Allgemeine Unfallversicherungsanstalt (AUVA)", "Jubiläumsfonds der Oesterreichischen Nationalbank", project 6774 and by the "Steiermärkische Landesregierung".

Abstract

The Electroencephalogram (EEG) is modified by motor imagery and can be used by patients with severe motor impairments (e.g. late stage of Amyotrophic Lateral Sclerosis) to communicate with their environment. Such a direct connection between the brain and the computer is known as an EEG-based brain-computer interface (BCI).

This paper describes a new type of BCI system that uses Rapid Prototyping to enable a fast transition of various types of parameter estimation and classification algorithms to real-time implementation and testing. Rapid Prototyping is possible by using Matlab, Simulink and the Real-time Workshop. It is shown how to automate real-time experiments and perform the interplay between on-line experiments and off-line analysis. The system is able to process multiple EEG channels on-line and operates under Windows 95 in real-time on a standard PC without an additional digital signal processor (DSP) board. The BCI can be controlled over the Internet, LAN or modem.

This BCI was tested on 3 subjects whose task it was to imagine either left or right hand movement. A classification accuracy between 70 and 95 % could be achieved with two EEG channels after some sessions with feedback using an adaptive autoregressive (AAR) model and linear discriminant analysis (LDA).

I. INTRODUCTION

An Electroencephalogram-based brain-computer interface (EEG-based BCI) provides a new communication channel between the human brain and the computer [1, 2, 3]. At this time, different brain signals can be used as input to a BCI: evoked potentials [4], slow cortical potentials (SCP) [5] or oscillatory EEG components [3, 6]. Patients who suffer from severe motor impairments (e.g. late stage of Amyotrophic Lateral Sclerosis) could use such a BCI system as an alternative communication channel through thoughts, whereby such an interaction between brain and computer must be realized in real-time [5]. Possible applications of an EEG-based BCI are e.g. to move a cursor by mental control, which allows the patient to select letters or words [3, 7], and to control a Functional Electrical Stimulation device for patients with spinal cord lesions [8]. These applications can be controlled by at least one binary output signal of the BCI, which is obtained, for example, by classification of EEG-patterns during imagination of left and right hand movements.

The use of oscillatory EEG components as input signals for a BCI requires on-line analysis of EEG signals with the extraction of reliable parameters. It was shown recently that unilateral hand movement imagery results in a contralateral event-related desynchronization (ERD) close to primary motor areas and, in certain cases, in conjunction with an ipsilateral event-related synchronization (ERS) of sensorimotor rhythms [9, 10]. A minimum of EEG channels is therefore assembled close to primary hand areas (electrode positions C3 and C4 according to the international 10-20 system) as an array of electrodes overlying motor and somatosensory areas. A recent paper based on discrimination of the EEG recorded during preparation for movement gives strong evidence that by the use of multiple EEG channels and common spatial filters, the classification accuracy of single-trial EEG can be increased [11].

In the BCI previously used in our laboratory ("old Graz-BCI"), the experimental paradigms were programmed in C and installed under MS-DOS. The system used a digital signal processor (DSP) board in addition to a PC [1] to handle the data acquisition (DAQ) and the analysis of EEG-data in real-time. This dependency on a DSP board increased the complexity of the BCI system, because DSP boards are usually programmed in C or assembly languages and therefore are harder to maintain or expand. A BCI system with a DSP board and a specially designed corresponding DAQ

board is more expensive, and therefore less suitable, especially for assistive technology applications. Another disadvantage of the old Graz-BCI was the necessary data conversion from the format used on the BCI system to Matlab where the off-line analysis was made. Routines for signal processing were written in Matlab (MathWorks, Inc., Natick, USA) and tested off-line on a PC prior to the use on the DSP board, but thereafter it was necessary to implement the routines in C to be suitable for the DSP of the BCI system. Therefore, it was decided to design and implement a new BCI that does not have these limitations. Moreover, an improved BCI that enables fast and easy implementation of different processing algorithms and classification methods is needed for optimal classification accuracy. A variety of different parameter estimation methods are now available for EEG processing: e.g. autoregressive spectral estimation [12], calculation of band power values in predefined frequency bands [1], calculation of adaptive autoregressive (AAR) parameters [13], usage of common spatial patterns [11], slow cortical potentials [5]. Further, different strategies of classification can be used: e.g. a linear threshold [5, 12], neural network classifiers [14] or a linear discriminant [15]. Ideally, a modern BCI system should be able to apply and compare all these methods and different combinations of them easily.

The main goal of this paper is to introduce the “new Graz-BCI” system that meets the following requirements:

- Works in real-time under Windows without a DSP board,
- Facilitates real-time implementation of EEG-analysis algorithms using Rapid Prototyping software (e.g. Matlab, Simulink and the Real-Time Workshop (MathWorks, Inc., Natick, USA)),
- Provides on-line display of EEG signals and stores all data for off-line analysis,
- Facilitates software maintenance and changes for different experimental paradigms,
- Provides fast feedback,
- Can be implemented on a laptop or wearable computer,
- Provides for on-line analysis of 2 up to 96 channels,
- Can be remotely controlled,
- Does not require format conversion for on-line and off-line analysis,
- Uses the same code for on-line and off-line analysis algorithms.

The requirement for Rapid Prototyping is especially important for determining optimal processing and classification algorithms because it enables rapid iteration of the design. This process is illustrated in the research documented in the present paper. Using the new Graz-BCI system, tests were conducted with 3 subjects to collect preliminary data on the following topics.

One research topic concerns whether it is possible to classify EEG data for on-line BCI operation by using both a recursive least squares (RLS) algorithm to estimate adaptive autoregressive (AAR) coefficients and a linear discriminant analysis (LDA) algorithm. A combination of these two methods in off-line analysis has been found to improve the classification accuracy of single EEG trials, in comparison to band power estimations and neural network-based classifiers [16]. The present effort examines whether these algorithms are suitable for on-line experiments.

A second topic pertains to the feedback provided to the operator. In previous studies using discrete feedback presented with delay [9, 16], classification accuracy did not increase over sessions. Delayed feedback can even degrade performance. A general enhancement of performance can be achieved using continuous feedback which is updated every 250 ms [17]. In the cited study, subjects learned through feedback control to enhance the left-right asymmetry with imagined unilateral hand movement. After a period of several sessions left-right EEG differences could be attained and classified in single-trial EEG data. Therefore, the present study's feedback condition employs a paradigm with continuous and immediate feedback. Performance with this condition is compared to a condition where no feedback was provided. Learning effects are also examined across each subject's sessions.

A third research topic pertains to determining whether updating the subject-specific weight vector after every 3 sessions improves classification accuracy, as has been suggested by [16]. The present study examines the performance effects of updating the classifier.

Before addressing each of these research topics, the design of the new Graz-BCI system will be described.

II. DESIGN OF THE NEW GRAZ-BCI SYSTEM

A. Hardware

The new Graz-BCI consists of an IBM compatible Pentium II PC operating at 233 MHz and an RTI800a data acquisition board (Analog Device, Norwood, USA), with 32 single ended analog input channels, 8 digital inputs and 8 digital outputs. Given the present system has 3 ISA slots, a maximum of 96 analog channels can be used. The analog to digital converter (ADC) has a resolution of 12 bits. For further hardware details see [18].

B. Operating System and Software Programming Method

In order to control the hardware, it is important to have a powerful software package that also includes the driver for the DAQ-board. Further, it is important to support new standards of operating systems. At present, Windows is the de facto desktop standard and a tremendous amount of software is available. However, Windows is not designed as a real-time operating system. Therefore, the PC, with Windows 95 installed, was equipped with a real-time Kernel expansion.

It is important to choose a programming language that enables an easy setup or adaptation of the programs for the experimental paradigm, data acquisition and signal analysis. For this purpose Matlab, in combination with the signalflow oriented Simulink Toolbox, was chosen. The system can be programmed graphically and it runs in real-time under Windows. The system is able to read and classify EEG data sample by sample with a maximum rate of 66 kHz without buffering on the DAQ-board and without using a DSP board for the real-time calculation. Thus, if 100 channels are used (software supports up to 256), each channel can be sampled at 660 Hz. The entire processing and control of data acquisition can be done by a single PC.

There is always a tradeoff between ease of programming with high level languages versus speed and flexibility with low level languages. Matlab routines are developed by professional programmers and are highly optimized, but time critical calculations can also be written in C and incorporated in Matlab programs. Simulink blocks can be written in C to improve the speed (C-Mex S-Functions). This was important because suitable algorithms for a BCI system were not available from Matlab or in a Simulink block library and had to be implemented by one of the authors [19]. Simulink is used for the calculation of different parameters, which describe the current state of the EEG in real-time (after real-

time code generation), while Matlab handles the data acquisition, timing and presentation of the experimental paradigm.

A further advantage of the new Graz-BCI is that it can be remote controlled over an analog dial-up, LAN or Internet connection. Details will be published in a separate paper.

C. The Development Process of the new Graz-BCI with Rapid Prototyping

Step 1: Selection of parameter estimation and classification algorithms

An appropriate method for an EEG-based BCI is e.g. the adaptive autoregressive parameter estimation. An AAR model describes a signal x_k in the following form:

$$\mathbf{x}^k = \mathbf{a}_{1,k}\mathbf{x}^{k-1} + \dots + \mathbf{a}_{p,k}\mathbf{x}^{k-p} + \varepsilon_k$$

where, in the ideal case, ε_k is a purely random or white noise process with zero mean and variance δ_ε^2 . The model is of order p and $a_{1,k} \dots a_{p,k}$ are the time varying AR-coefficients, which are estimated with the recursive least squares (RLS) algorithm. For a detailed description see [13, 20].

Linear discriminant analysis [21] is used to differentiate between EEG patterns associated with left and right movement imagery (see also Section III).

Step 2: For analyzing the EEG data, Matlab and Simulink are used. Algorithms can be programmed in Matlab or a block-diagram can be developed using Simulink. Simulink is an interactive environment for modeling, analyzing and testing dynamic systems and runs under Matlab. Simulink's graphical user interface enables the user to build block diagrams using drag-and-drop techniques and provides the possibility to write special blocks for on-line analysis, called S-functions (System-functions) [22].

Common spatial filters [11] and the following parameter estimation algorithms [19] have been implemented in the new BCI system: Hjorth [23], Barlow [24], RLS [20], least mean squares (LMS) [20], fast Fourier transformation (FFT) and band power calculation [1]. S-function blocks, which use Matlab code, have to be rewritten into C-Mex S-functions (S-Functions programmed in C) if real-time code should be generated (see Step 5) with the Real-Time Workshop [22]. After compiling C-Mex S-functions as Matlab-Executable-files (Mex-files) they can be used in Simulink more than once because of their reentrancy feature achieved with the SimStruct (Simulation Structure) data structure [22].

The AAR-model was implemented with Simulink and C-Mex S-functions [19] as shown in Figure 1. A model order of $p=6$ and an update coefficient of 0.006 was used [13]. The algorithm was initialized at

the beginning of every trial to avoid instabilities [19]. Fisher's linear discriminant analysis was implemented in Matlab for off-line analysis [15].

Figure 1

Step 3: Off-line simulations and tests of the Simulink block diagrams and the implemented S-function blocks were performed with previously recorded EEG-data of former BCI-experiments [16]. Usually steps 2 and 3 are repeated many times to improve the block model and the implementation of the S-functions and to reduce development work in future steps.

Step 4: If no further improvements can be made with off-line tests then the model must be connected to the real world. Real-time programs communicate with external input/output (I/O) devices via a device driver that contains the necessary code to interface Simulink to the RTI800a DAQ board. The device driver is configured via a dialog box where the address I/O space of the board, the hardware gain (1, 10, 100, 500), the number of analog input channels and the sampling frequency must be specified. Next, the 'Read data' blocks are replaced using drag-and-drop by an input block (analog to digital converter) as shown in Figure 2.

Figure 2

Step 5: The key advantage of Rapid Prototyping is the automatic code generation, which provides a means to create programs for real-time applications. Once the desired results are achieved with Simulink off-line tests (Step 3), the real-time C code is directly generated, compiled, linked and downloaded to a real-time Kernel with the Real-Time Workshop (RTW) as shown in Figure 3. The C code of the C-Mex S-functions is automatically incorporated, but to save time it is also possible to incorporate S-functions directly into the generated code of the RTW so that no function-calls to the S-function itself are necessary [25]. The RTW enables the user to quickly run a Simulink design in real-time without hand programming and debugging. The automatic code generation, even for complex systems, normally takes only some minutes.

Figure 3

Step 6: Simulink in the External Mode can be used as a graphical front-end to the corresponding model. When the model is downloaded to the Kernel it can be started from Simulink and runs in real-time under Windows. An Interprocess Communication Channel connects the real-time process (server) to the Simulink block diagram (client). Simulink waits for any parameter change and sends

changed ones immediately to the running real-time program. With the Matlab Application Program Interface (Matlab API) and the Simulink External Interface (see Figure 3) it is possible to interact with the real-time program without stopping the execution. During the experiments, all data from the EEG channels are recorded and this data can be used for re-modeling and re-designing the Simulink model after off-line analysis in Matlab and/or Simulink. Since off-line and on-line analyses are performed in the same environment, alternative signal analysis methods, suggested by off-line analysis in Step 1 to Step 3, can be incorporated into on-line experiments quickly. This Rapid Prototyping enables the real-time system to be refined by continuously iterating the model quickly and easily.

Step 7: The real-time tests give information about where the model must be improved to reduce noise or disturbances, correct timing, improve the algorithm robustness to artifact (e.g. eye blinks) or increase data storage reserved for data logging. Sometimes it might be necessary to go back to Step 1 to adjust the algorithm.

III. EXPERIMENT IMPLEMENTATION USING MATLAB

Three male, right handed students (S1, S2, S3) of age 16 to 25 were tested on the new Graz-BCI. The students were paid per session and were free of medication and central nervous system abnormality. S1 already participated in a number of experiments [16, 17] with the old Graz-BCI [1]. This was the first BCI experiment for S2 and S3.

During each experimental session, the real-time program communicates with the Host PC for parameter exchange, data acquisition and on-line display. The functions of the Matlab API give access to parameters of the Simulink model and to the inputs and outputs of the DAQ board without interrupting the real-time operation. The following provides specific information about data recording, experimental paradigm, and classifier setting.

A. Data Recording

The Matlab API enables an automatic capture sequence to run under the control of Matlab. In the Matlab program, inputs, outputs and various parameters, such as the subject-specific weight vector (see Section III.C), the gain (see Figure 2) or the sampling rate, can be specified for capture.

Two bipolar EEG-channels were recorded over left and right sensorimotor hand areas, close to electrode positions C3 and C4. The EEG was amplified by an 8-channel amplifier system, sampled at 128 Hz and bandpass filtered between 0.5 and 30 Hz. Artifacts, such as eye movements, were not

discarded for on-line EEG analysis. EOG (Electrooculogram) was derived from an electrode placed medially just above the right eye and a second electrode laterally just below the right eye to detect vertical as well as horizontal eye movements. The surface EMG (Electromyogram) was recorded from the extensor muscles of both hands. Both, EMG and EOG recordings, were made during the last session for each subject to ensure that the EEG-based BCI system was not controlled by eye movements or muscle activity.

While performing the presented experimental paradigm (see Figure 4) the data from 2 EEG, 1 EOG, 2 EMG channels and 1 trigger signal were logged from the Simulink real-time application into the random access memory (RAM) of the PC. After the experiment, the data were transferred directly to Matlab. The data were stored in Matlab format, but an ASCII or binary format can also be selected. This acquired data allows comprehensive off-line analysis in the same Matlab environment. If the system is equipped with 128 megabyte RAM and acquires 10 channels at 100 Hz it can store approximately 17 hours of EEG data.

B. Experimental Paradigm

To demonstrate the interaction between Matlab and the real-time model we used a paradigm that gives fast and continuous feedback by using Matlab's numerical computation and graphics capabilities (see Figure 4). The function for reading the classification result of the Simulink model of Figure 2 into Matlab needs 2 ms. The classification result is then converted to a horizontal feedback bar and displayed within 20 ms. Therefore, the bar can be updated at approximately 50 Hz, which is sufficient for human perception.

The synchronization between the real-time process and the Matlab program is also made with the API with an accuracy of 200 μ s. The communication channels allow ongoing parameter adjustments, e.g. if the amplitude of the feedback bar reaches the edge of the screen it can automatically be made shorter by considering that in the Matlab program, or manually, on the fly, by changing the value of the gain block in Figure 2.

Figure 4

C. Classifier Setting

The tight coupling between the on-line experiments and off-line analysis of the acquired data is one of the major advantages of the new Graz-BCI system, particularly for classifier setting. There were two types of recording sessions: in one type, data were collected to establish a subject-specific weight

vector (classifier in Figure 4) and in the other type, the subject-specific weight vector was used to classify the EEG on-line while the subject imagined the requested kind of movement.

In session 1, the paradigm that is described in Figure 4, but without feedback, was performed to obtain this subject-specific weight vector. After the session, the acquired data can be applied off-line to the Simulink model that is shown in Figure 1, to calculate the AAR-coefficients with the RLS algorithm for every iteration. The AAR-coefficients at 8 different time points were stored to harddisk with the 'Write' block. To obtain a more general view of the classification ability, a 10 times 10 fold cross validation of a linear discriminant was performed off-line for each of the eight classification time points [21]. The 10 times 10 fold cross validation mixes the data set randomly and divides it into 10 equally sized disjunct partitions. Each partition is then used once for testing, the other partitions are used for training. This results in 10 different error rates, which are averaged. This is the error rate of a 10 fold cross validation. To further improve the estimate the procedure is repeated 10 times and again all error rates are averaged.

The AAR-coefficients of the classification time points with the lowest classification error were used to set up the subject-specific weight vectors with the LDA for the following sessions with feedback (see Figure 5) [16, 26]. This off-line procedure, from reading the recorded data from harddisk until the availability of the new weight vector, requires approximately 5 minutes and enables the next session to start immediately after calculation. The classification result can continuously be viewed with the Scope Classification Result (see Figure 2) during real-time operation.

In sessions 2 and 3 the outputs of the RLS-algorithms were calculated and classified with the weight vector in real-time to show the feedback on-line in form of a bar on the screen. The bar, varying in length, pointed to the left if the output of the linear classification was positive and to the right if negative. The size of the bar was determined by the absolute value of the classification result, a measure of how reliably the side was determined.

Figure 5

A new weight vector was established from the data recorded in session 4, again without feedback (Figure 5, right side). This weight vector 2 was used in sessions 5 and 6 to give the feedback. Subject S2 participated in a seventh session and therefore the data of session 6 were used to make a third weight vector. Weight vector 3 was used in session 7 to give feedback to S2.

IV. RESULTS

Classification results obtained with subjects S1, S2 and S3 are graphically presented in Figure 6. Altogether 19 sessions, including 160 trials each, were held. Sessions 1 and 4 were performed without feedback, where the subjects were instructed to imagine a right or left movement after the cue presentation. Sessions 2, 3, 5, 6 and 7 were performed with feedback, as described in Section III. The feedback (horizontal bar) was displayed on-line on the monitor from second 4.25 until second 8 and was continuously updated in real-time with the subject-specific weight vector obtained in sessions 1, 4 and 6, respectively. In the feedback sessions the classification error ranged from 5 to 29.4 % for all classification time points and subjects. The lowest error rate for S1 (5 %) was observed at second 6, the lowest error rate for S2 (9.8 %) at second 7 and for S3 (9 %) at second 6. It is of interest to note that the similar results obtained in sessions 2, 3, 5, 6 and 7 (all with feedback) are clearly different from sessions 1 and 4 (without feedback), except session 4 of S3. It is also important to note that the minimum error rate decreased from 16.9 % in the first feedback session to 9.8 % in the last session of subject S2. The same trend can be seen with subject S3; the minimum error rate decreased from 14.75 % in the first feedback session to 9 % in the last session. With S1 the minimum error rate occurred in the first feedback session.

Figure 6

The results obtained from the 3 subjects show basic differences:

- Subject S1 already participated in a number of BCI-experiments with our old Graz-BCI system [1]. This might explain why the error rate did not improve across sessions with feedback and ranged from 5 to 18.9 %. Generally, error rate was lower in sessions with feedback. The update of the classifier after session 4 had nearly no effect on the error rate. It is also of interest to note that the error rate increased in the sessions without feedback with increasing classification time point to about 40 %; only shortly after the cue-presentation the error rate was around 20 %. Subject S1 reported a better controllability of the feedback bar when it is presented immediately (20 ms) compared to a delay of 250 ms [17]. The last two sessions were performed with the remote control over a LAN.
- Subject S2 showed an error between 32 % and 47 % in the first session without feedback. In feedback session 2 the minimum error rate decreased to 16.9 % and in feedback session 3 to 17.8 % (time point 5.75 seconds). After an update of the weight vector with the data of session 4 and later of session 6, the minimum error rate was reduced further to 9.8 % (time point 7

seconds). The training and update of the weight vector yielded an improvement of the classification results as suggested in [16]. Subject S2 had never participated in BCI-experiments before and was able to reach a minimum error rate of 9.8 % within only 7 sessions.

- Subject S3 showed an error rate between 21.3 and 49.4 % in the first session. In the following feedback sessions the minimum error decreased to 14.75 % (session 2 at second 5.5) and to 10.6 % (session 3 at second 7). In session 5 the error rate increased slightly. In the last session the subject achieved the lowest error rate (9 %) of all sessions. The update of the weight vector in session 4 did not show a significant effect on the error rate.

V. DISCUSSION

A. Design of the new Graz-BCI

This paper describes the development of a BCI system with Rapid Prototyping. Model design, real-time implementation, experiments and off-line analysis are performed in one environment and this allows a very efficient project development. The engineer can concentrate on the project and not on tool-development, which results in a shortened development process. Developers don't lose time with low-level programming and the creativity is increased by an open architecture.

The multichannel system runs under Windows 95 and is equipped with a real-time Kernel expansion to obtain reasonable real-time operations on a standard PC or laptop computer. Matlab handles the data acquisition and the presentation of the experimental paradigms while the signal flow oriented Simulink is used to describe the current state of the EEG on-line.

The autoregressive parameter estimation has the advantage that no reactive frequency bands must be specified a priori as necessary for band power values [1]. The AR parameters are estimated from the entire EEG, limited only by the lower and upper cutoff frequencies of the EEG amplifier. The tight coupling between on-line experiments and off-line analysis allows to set up a classifier within 5 minutes and to start the next session immediately after calculation. Another advantage is that the classifier can be set up with a relatively small number of training trials (160 trials).

The new Graz-BCI is equipped with a remote control that allows the control of the system over an analog dial up, LAN or Internet connection from a different room (enhancing laboratory testing) or a different geographic location. The latter capability greatly facilitates application of BCI systems in assistive technology [27]. The patient's system can be remotely updated, to change the classifier or

analysis method or install improved software. Similarly, acquired EEG data can be transmitted to the BCI developer for off-line processing.

B. Experiment Conducted with new Graz-BCI

The experiment demonstrated, for the first time, that the RLS and LDA algorithms can be used in a BCI system to calculate feedback in real-time. Within a small number of sessions (even in naive subjects) classification accuracy improved with the continuous feedback representing motor imagery-related EEG changes.

All subjects showed the lowest classification results for the feedback sessions in the middle classification time points, with larger error in the early and late time points.

Moreover, error rate was generally lower in the sessions providing feedback compared to sessions with no feedback. The classification improved for 2 of the 3 subjects across feedback sessions.

Performance effects from updating the classifier were less clear. For two of the subjects (S1 and S3), the classifier update did not appear to affect the error rate. For subject S2, the classifier update appeared to decrease the error rate. Further experimentation is required to address in more detail the role of feedback and classifier adjustment in the successful learning and operation of this BCI approach. Additional testing is also needed to confirm that the successful performance results from the three subjects in this preliminary study hold true for a larger sample. The preliminary results from the present study do, however, suggest that operation and application of the new Graz-BCI are very promising.

VI. CONCLUSION

The presented new Graz-BCI can be used as an experimental system and assistive tool at the patient's home as well. By using two bipolar EEG channels with the RLS and the LDA algorithms a high classification accuracy can be achieved after only a few sessions. Rapid Prototyping allows specific analysis methods to be updated very quickly and keeps the system very flexible. Assistive technology applications of this approach to BCI system design are promising given that the modular design can be installed on a laptop computer and system maintenance can be accomplished by remote control.

ACKNOWLEDGEMENTS

We thank Gloria Calhoun for the excellent assistance. Further we like to thank Britta Ortmayr, Bernhard Obermaier, Jimmy Demon and Peter Ankarson for critically reading the article and for the valuable advice.

REFERENCES

- [1] J. Kalcher, D. Flotzinger, C. Neuper, S. Göllly, and G. Pfurtscheller, "Graz Brain-Computer Interface II: towards communication between humans and computers based on online classification of three different EEG patterns," *Med. Biol. Eng. Comput.*, vol. 34, pp. 382-388, 1996.
- [2] J.J. Vidal, "Toward direct brain-computer communication," *Annu. Rev. of Biophys. Bioeng.*, pp. 157-180, 1973.
- [3] J.R. Wolpaw and D.J. McFarland, "Multichannel EEG-based brain-computer communication," *Electroenceph. Clin. Neurophysiol.*, vol. 90, pp. 444-449, 1994.
- [4] E. Sutter, "The brain response interface: communication through visually-induced electrical brain response," *J. Microcomput. Applications*, vol. 15, pp. 31-45, 1992.
- [5] A. Kübler, B. Kotchoubey, T. Hinterberger, N. Ghanayim, J. Perelmouter, M. Schauer, C. Fritsch, E. Taub, and N. Birbaumer, "Thought Translation Device: A Neurophysiological Approach to Communication in Total Motor Paralysis," *Exp. Brain Research*, vol. 124, pp. 223-232, 1999.
- [6] G. Pfurtscheller, J. Kalcher, C. Neuper, D. Flotzinger, and M. Pregenzer, "On-line EEG classification during externally-paced hand movements using a neural network-based classifier," *Electroenceph. Clin. Neurophysiol.*, vol. 99, pp. 416-425, 1996.
- [7] N. Birbaumer, N. Ghanayim, T. Hinterberger, I. Iversen, B. Kotchoubey, A. Kübler, J. Perelmouter, E. Taub, and H. Flor, "A spelling device for the paralysed," *Nature*, vol. 398, pp. 297- 298, 1999.
- [8] D.B. Popovic, R.B. Stein, K.L. Jovanovic, R. Dai, A. Kostov, and W.W. Armstrong, "Sensory nerve recording for closed-loop control to restore motor functions," *IEEE Biomed. Eng.*, vol. 40(10), pp. 1024-1031, 1993.

- [9] G. Pfurtscheller, C. Neuper, D. Flotzinger, and M. Pregenzer, "EEG-based discrimination between imagination of right and left hand movement," *Electroenceph. Clin. Neurophysiol.*, vol. 103, pp. 642-651, 1997.
- [10] G. Pfurtscheller and C. Neuper, "Motor imagery activates primary sensorimotor area in humans," *Neurosci. Lett.*, vol. 239, pp. 65-68, 1997.
- [11] J. Müller-Gerking, G. Pfurtscheller, and H. Flyvbjerg, "Designing optimal spatial filters for single-trial EEG classification in a movement task," *Electroenceph. Clin. Neurophysiol.*, vol. 110, pp. 787-798, 1999.
- [12] D.J. McFarland, A.T. Lefkowitz, and J.R. Wolpaw, "Design and operation of an EEG-based brain-computer interface (BCI) with digital signal processing technology," *Behav. Res. Meth. Instr. Comput.*, vol. 29, pp. 337-345, 1997.
- [13] A. Schlögl, D. Flotzinger, and G. Pfurtscheller, "Adaptive autoregressive modeling used for single-trial EEG classification," *Biomed. Technik*, vol. 42, pp. 162-167, 1997.
- [14] D. Flotzinger, J. Kalcher, and G. Pfurtscheller, "EEG classification by learning vector quantization," *Biomed. Technik*, vol. 37, pp. 303-309, 1992.
- [15] K. Lugger, D. Flotzinger, A. Schlögl, M. Pregenzer, and G. Pfurtscheller, "Feature extraction for on-line EEG classification using principal components and linear discriminants," *Med. Biol. Eng. Comput.*, vol. 36, pp. 309-314, 1998.
- [16] G. Pfurtscheller, C. Neuper, A. Schlögl, and K. Lugger, "Separability of EEG signals recorded during right and left motor imagery using adaptive autoregressive parameters," *IEEE Trans. Rehab. Engng.*, vol. 6, pp. 316-325, 1998.
- [17] C. Neuper, A. Schlögl, and G. Pfurtscheller, "Enhancement of left-right sensorimotor EEG differences during feedback-regulated motor imagery," *J. Clin. Neurophysiol.*, vol. 16(2), 1999.

- [18] C. Guger, A. Schlögl, D. Walterspacher, and G. Pfurtscheller, "Design of an EEG-based brain-computer interface (BCI) from standard components running in real-time under Windows," *Biomed. Technik*, vol. 44, pp. 12-16, 1999.
- [19] C. Guger, "Implementation of different EEG processing algorithms with S-Functions," Master thesis, University of Technology Graz, 1997.
- [20] S. Haykin, *Adaptive Filter Theory*. New Jersey: Prentice Hall, Englewood Cliffs, 1986.
- [21] C.M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Clarendon, 1995.
- [22] The MathWorks Inc, *Simulink. Dynamic System Simulation for Matlab*. Natick, MA: The MathWorks, 1997.
- [23] B. Hjorth, "EEG analysis based on time domain properties," *Electroenceph. Clin. Neurophysiol.*, vol. 29, pp. 306-310, 1970.
- [24] I.I., Goncharova, J.S. Barlow, "Changes in EEG mean frequency and spectral purity during spontaneous alpha blocking," *Electroenceph. Clin. Neurophysiol.*, vol. 76, pp. 197-204, 1990.
- [25] The MathWorks Inc, *Simulink. Real-Time Workshop*. Natick, MA: The MathWorks, 1997.
- [26] A. Schlögl, K. Lugger, and G. Pfurtscheller, "Using adaptive autoregressive parameters for a brain-computer-interface experiment," in *Proc. IEEE-EMBS'97*, pp. 1533-1535, 1997.
- [27] T.M. Vaughan, J.R. Wolpaw, and E. Donchin, "EEG-based communication: Prospects and problems," *IEEE Trans. Rehab. Engng.*, vol. 4, pp. 425-430, 1996.

LEGEND OF FIGURES

Figure 1: The 'Read data 1' and 'Read data 3' blocks read the EEG data previously recorded (from bipolar EEG channels close to C3 and C4 of the 10-20 system) from Matlab into the Simulink environment with a sampling frequency of 128 Hz. 'RLS channel #1' and 'RLS channel #2' calculate the RLS-algorithm with a sampling rate inherited from the blocks driving them (128 Hz). The output of the RLS-algorithms consists of six ($p=6$) time varying AR-coefficients for each EEG channel. 'Read data 2' reads a 'Trigger' signal that indicates the starting time point of the beep of each trial (see Figure 4). The 'Write' block stores the outputs of the RLS-algorithms to harddisk at a specific time point indicated by a trigger signal on input 3 of the multiplexer. The 'Delay' block delays passing of the 'Trigger' signal to the 'Write' block. This allows a specification of a time point which indicates when the outputs of the RLS-algorithms (input 1 and 2 of the multiplexer) should be written to harddisk. Then the AAR-coefficients can be applied off-line to Fisher's linear discriminant analysis (LDA) to set up a subject-specific weight vector for the experiments with feedback as described in Section III. The Scopes are used to display the outputs of the blocks on-line.

Figure 2: A device driver for the RTI800a (DAQ board from Analog Device) is added to the model to make the connection to the real world. In this case the input block represents analog input channel 1 to 3 (EEG#1, EEG#2, Trigger). Channel 1 and 2 are connected to the 'RLS+LDA' algorithm blocks and channel 3 is the trigger signal used to initialize the RLS-algorithms at the beginning of every trial. The output of the RLS-algorithm is classified with a weight vector previously obtained from a linear discriminant analysis (LDA). The on-line classification result is amplified and displayed with the Scope block 'Classification Result'.

Figure 3: After designing the Simulink model, C code is generated, compiled, linked and downloaded to the real-time (RT) Kernel. The External Mode of Simulink enables the user to view outputs of Simulink blocks on-line on digital oscilloscopes and to tune parameters on-line (e.g. the weight vector or the gain of the model in Figure 2 can be changed on the fly). Data from all inputs and outputs of the DAQ board can be read into Matlab via the Matlab Application Program Interface (API). This is required to update the feedback bar (see Section III) according to the real-time classification, store the

EEG data to harddisk or set software-switches (e.g. if an artifact is detected, switch off the feedback in this trial). Via DDE (Dynamic Data Exchange) it is possible to send data or classification results to other Windows based utilities.

Figure 4: Timing of one trial of the experiment with feedback. The feedback is in form of a horizontal bar. The subject sat in a comfortable armchair 150 cm in front of a computer screen and was instructed not to move and to keep both arms and hands relaxed. The experiment started with the display of a fixation cross that was shown in the center of a monitor. After two seconds a warning stimulus was given in form of a "beep". From second 3 until 4.25 an arrow (cue stimulus), pointing to the left or right, was shown on the screen. The subject was instructed to imagine a left or right hand movement until the end of the trial, depending on the direction of the arrow. The EEG was sampled and classified on-line throughout the session. Between second 4.25 and 8 the classification result was used to give a continuously updated feedback stimulus in form of a horizontal bar that appeared in the center of the screen. If the person imagined a left movement, then the bar, varying in length, extended to the left as shown in A and vice versa in B (correct classification assumed). The subjects' task was to extend the bar toward the left or the right boundary of the screen, during seconds 4.25-8.

One trial lasted 8 seconds and the time between two trials was randomized in a range of 0.5 to 2.5 seconds to avoid adaptation. Each session was divided into 4 experimental runs of 40 trials, with randomized direction of the cues (20 left and 20 right) and lasted about 1 hour (including electrode application, breaks between runs and all settings for the experiment). Six sessions were conducted with subjects S1 and S3 and seven with subject S2.

Figure 5: Session 1 was performed without feedback (FB) and the data were used to set up the subject-specific weight vector 1. In sessions 2 and 3 weight vector 1 was used to give feedback in the form of a bar. In session 4 (without feedback) a new weight vector 2 was established and used in sessions 5 and 6 to give the feedback. Only S2 participated in session 7 and therefore weight vector 3 was established from session 6 for just one subject, which is indicated by the dotted line. Subject S1: sessions 1,2 and 3 were performed within one week, sessions 4 and 5 eight weeks later and session 6 after another four weeks. Subject S2: sessions 1, 2 and 3 were performed within one week, sessions 4 and 5 two weeks later and sessions 6 and 7 again two weeks later. Subject S3: sessions 1 to 6 were performed within one week.

Figure 6: The error rate (100 % minus correct classification), obtained with a 10 times 10 fold cross validation of a linear discriminant, is displayed over classification time points and different sessions for subjects S1, S2 and S3 obtained with the operation of the new Graz-BCI system.

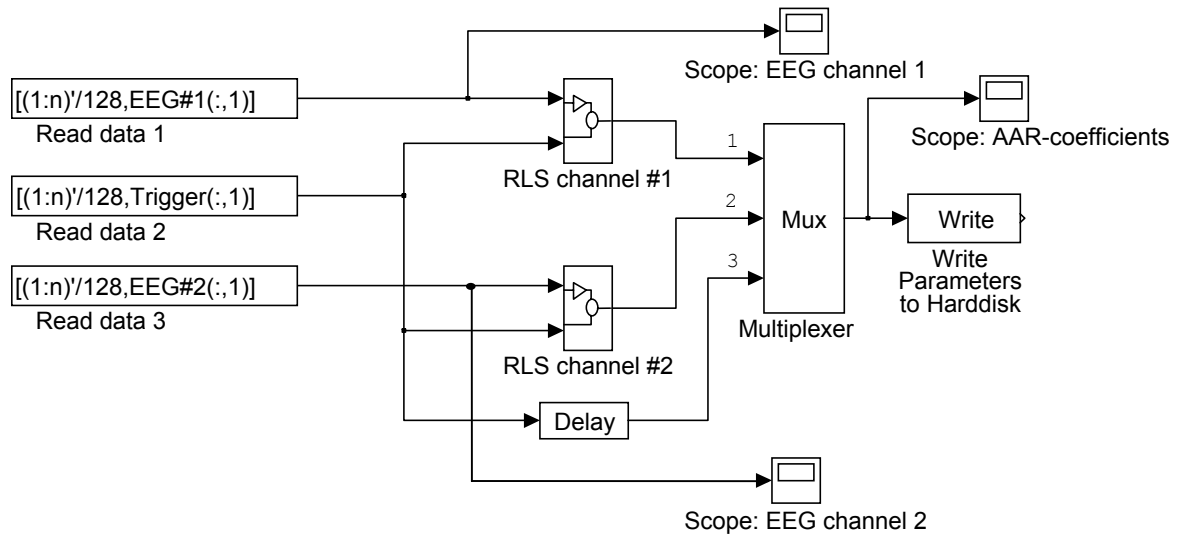


Figure 1

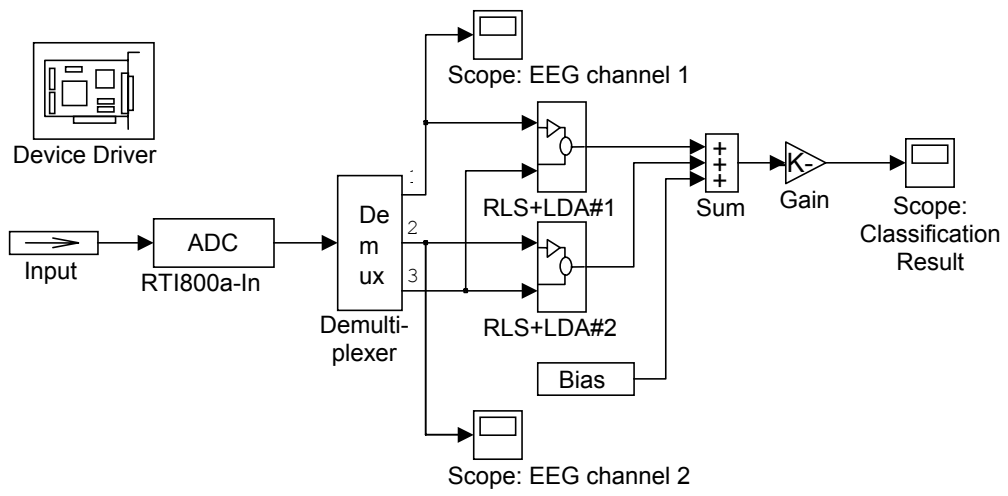


Figure 2

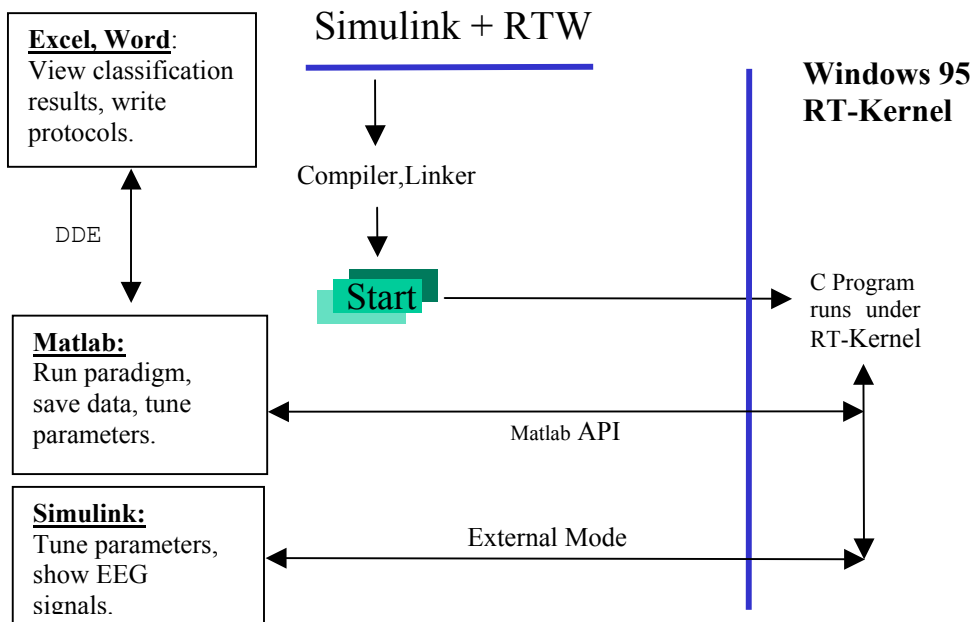


Figure 3

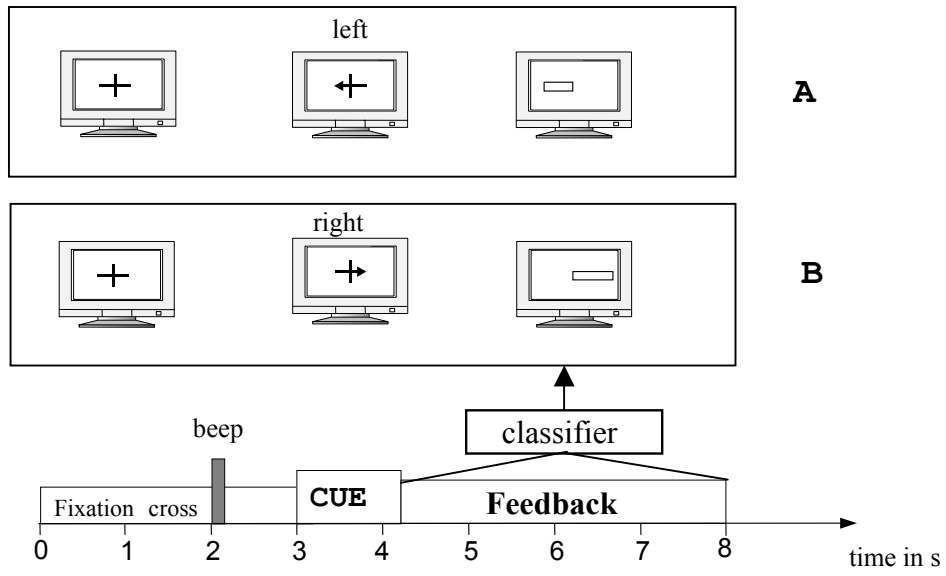


Figure 4

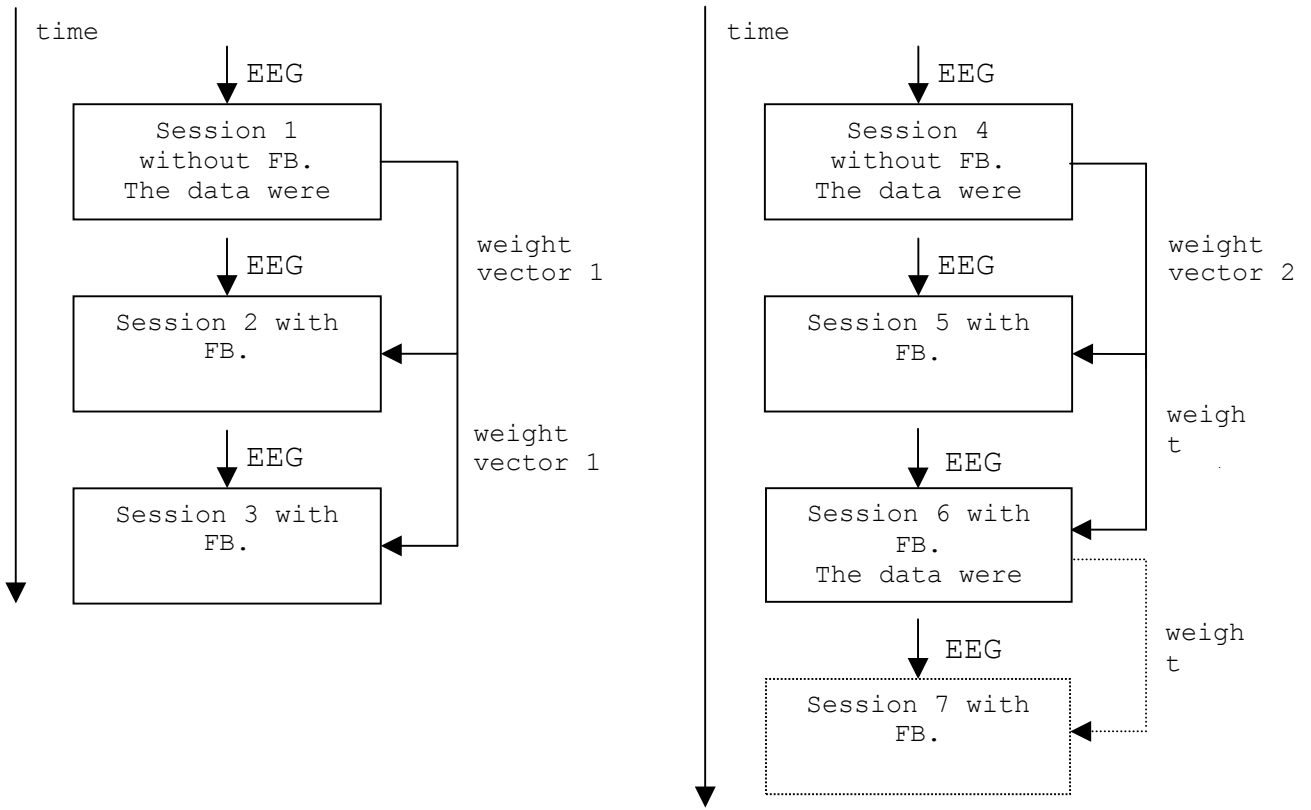


Figure 5

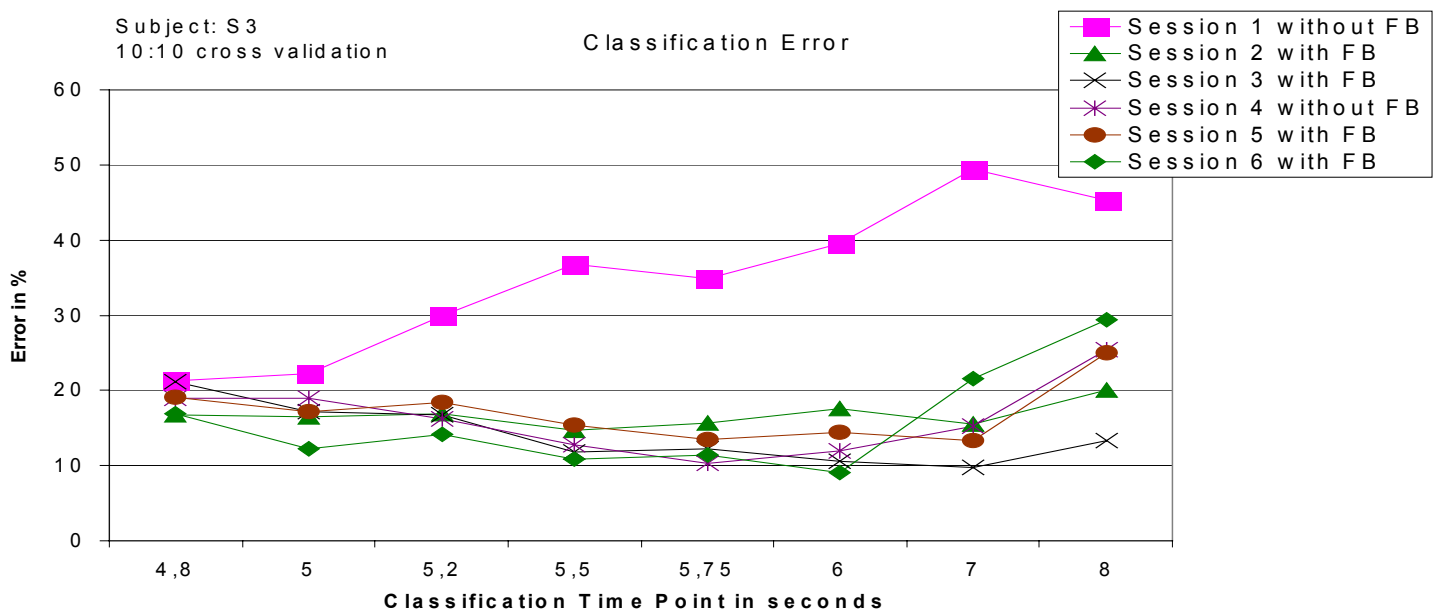
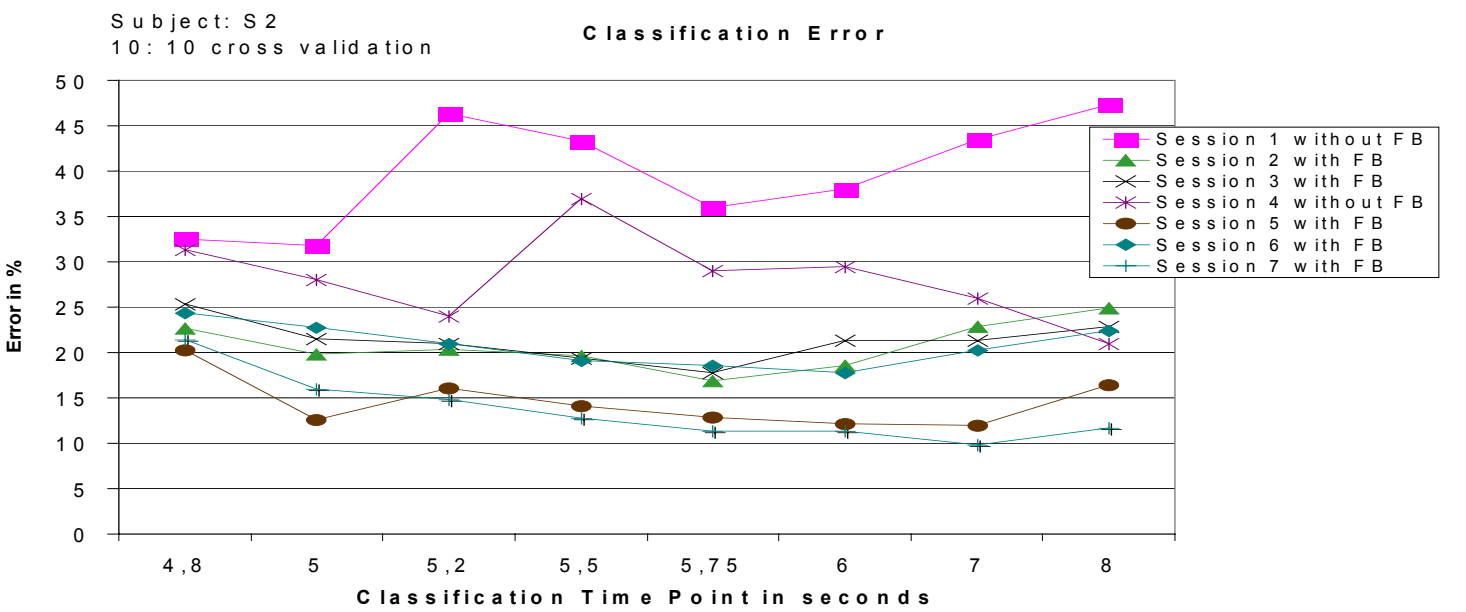
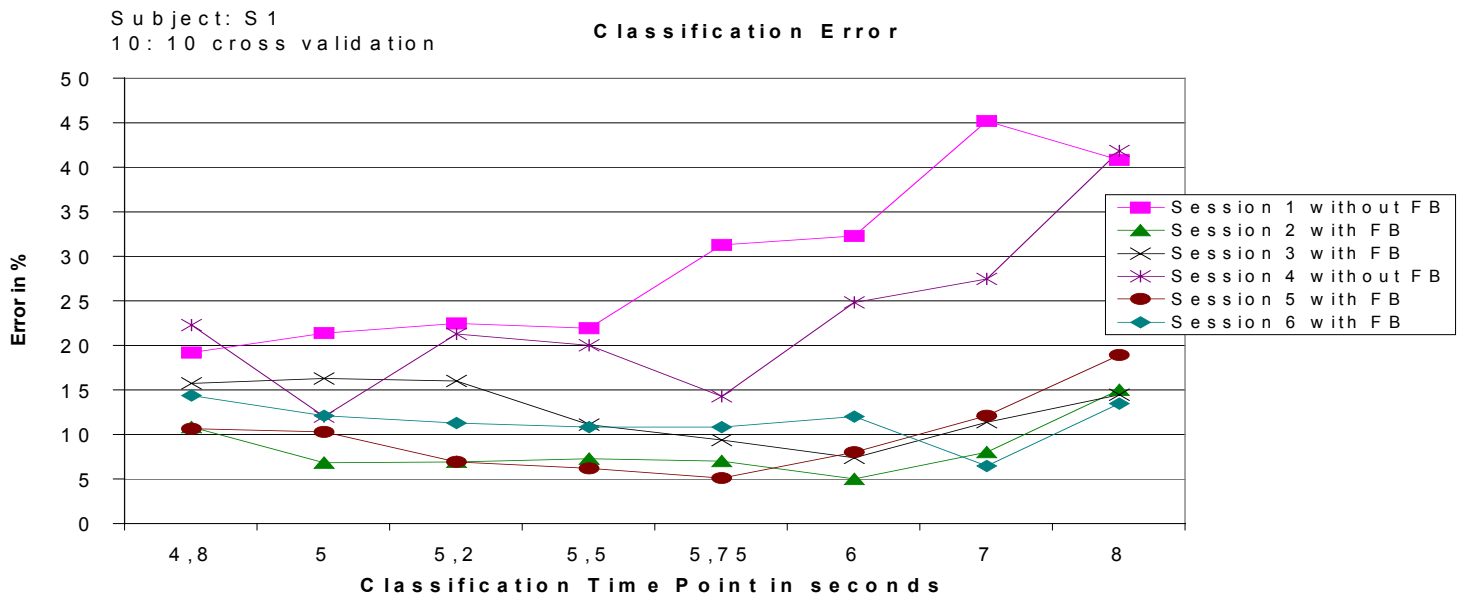


Figure 6