BCI using imaginary movements: The simulator

Darius A. Rohani a,*,1, William S. Henning a,1, Carsten E. Thomsen b, Troels W. Kjaer c, S. Puthusserypady a, Helge B.D. Sorensen a

a Technical University of Denmark, Department of Electrical Engineering, Denmark
b University of Copenhagen, Faculty of Health Science, Denmark
c Copenhagen University Hospital, Department of Clinical Neurophysiology, Denmark

ARTICLE INFO

Article history:
Received 13 January 2012
Received in revised form 4 February 2013
Accepted 17 April 2013

Keywords:
Brain Computer Interface
Communication rate
Hex-O-Spell interface
Parameter optimization
Sensory motor rhythms
Simulation

ABSTRACT

Over the past two decades, much progress has been made in the rapidly evolving field of Brain Computer Interface (BCI). This paper presents a novel concept: a BCI-simulator, which has been developed for the Hex-O-Spell interface, using the sensory motor rhythms (SMR) paradigm. With the simulator, it is possible to evaluate how the model parameters such as error classifications, delay between classifications and success rate affect the communication rate. Another advantage of the simulator is that it allows us to study for more classes than most online BCI systems which are limited to only two classes. Results show that the BCI simulator is able to give a deeper understanding of the feedback systems. We also find that a 3-class system is more efficient than a 2-class system if it obtains a success rate of at least 55% of the 2-class system.

© 2013 Elsevier Ireland Ltd. All rights reserved.

1. Introduction

The scientific field of Brain Computer Interface (BCI) works on the idea of interpreting brain signals, and converting it to outputs compatible with a computational device. Ever since the concept of BCI was introduced, there has been exponential growth in research groups all over the scientific community [1]. The primary reason for such high interest is that it helps improve the life quality of patients with severe neuromuscular damage where speech and movement of limbs is impaired or impossible. BCI systems enable such patients to communicate even without the output channels of peripheral nerves and muscles [2]. The communication can be in the form of a spelling program, neuroprostheses or other devices [3]. BCI is however limited by a low communication rate ranging from 5 to 84.7 bits/min [4]. Improvements in the signal processing algorithms, translation algorithms, signal acquisition, user training/adoptions, and control-interface are the key elements for increasing the communication rate [3,4]. Communication rate furthermore depends on the type of electrophysiological brain signals (sensory motor rhythms (SMR) [6–8], slow cortical potentials (SCP) [9], steady state evoked potentials (SSEP) [10], and P300 evoked potentials [11]) that is used in the systems [5]. The simulator proposed in this paper is based on the SMR paradigm, in which the subject imagines movement of extremities. This will commence a decrease in amplitude and desynchronization of the signals over motor cortex with...
frequency range of 8–30 Hz [6–8]. These changes in amplitude and desynchronization can be used in controlling BCI systems. The improvements of BCI feedback/control systems to increase the communication rate have been limited to online tests. In [20], the optimization with a Graz BCI based virtual keyboard paradigm is done in online mode with subjects being used in exhausting recording sessions. The procedure is very time-consuming, cumbersome as well as expensive. This motivated us to propose the BCI-simulator as a tool to overcome these factors by running simulations beforehand. It gives an estimation of whether changes are worth implementing in real BCI systems. In addition to improved feedback, the simulator helps us to better understand how a BCI feedback system works in general. The proposed BCI simulator thus helps us to improve the communication rate of today’s BCI systems by operating on the feedback system.

The simulator is based on a remodeled version of the Hex-O-Spell system [12,13] along with three training systems (Cross, Bars and Basket) [19]. No one has yet developed a simulator for a BCI feedback system which can be tested in offline mode. However the simulation term is not an unknown terminology in the BCI literature [14,17]. Reference [14] uses a mathematical simulation to investigate the possibility that a multi-class BCI system has accuracy at the same level as that of random classifications.

In this paper, we introduce the build-up of the simulator, with an explanation of the parameters implemented. Next, we show some results generated by the simulator with interest to verify it and use it to improve the communication rate. Finally, we discuss the results in detail with suggestions on future developments to make the simulator even more realistic and comprehensive.

2. Methodology

The proposed simulator is compatible with the Technical University of Denmark (DTU) version of the Hex-O-Spell [15] and the three training systems [19]. The DTU Hex-O-Spell is an expansion of its original version 1 developed by the BCI research group in Berlin (BBCI) [14,15].

2.1. Hex-O-Spell

The Hex-O-Spell is a 2-class BCI system based on the SMR paradigm. It is a spelling program, operating with an arrow rotating between 6 different letter-group hexagons. Class 1, which could be an imagined left-hand movement, is rotating the arrow clock-wise, while class 2, which could be an imagined foot movement, is elongating the arrow until a maximal length causing a selection to be performed. After selecting a letter-group, each hexagon is cleared, and the letters in the chosen group are sorted such that five hexagons now contain one letter while the last hexagon contains a return option. A subsequent round of rotation and selection is needed to pick a specific letter. The return option provides the user with the opportunity to get back to the letter-group state if the user chose a wrong group. Similarly if a user has selected the wrong letter, a delete symbol is available at the first hexagon letter-group.

The DTU version of the Hex-O-Spell has been modified to include a third imaginary movement to build the 3 class Hex-O-Spell BCI system. The third movement is used to shift between two Hex-O-Spell groups. The first group is the normal letter-picking Hex-O-Spell and the second is a dictionary Hex-O-Spell with the most likely words based on the present letters selected (see Fig. 1). It is however likely that the increased complexity will result in a decreased success rate. One of the main focuses of the simulator is therefore to investigate how large a reduction in success rate can be allowed for a 3-class system to still be more efficient than a 2-class system.

The motivation for modifying the original Hex-O-Spell is to achieve a higher communication rate, and thereby making it the preferable choice of communication interface. In order to make the BCI simulator realistic, an analysis is needed to decide on which input parameters are essential. The parameters for this simulator were chosen as Success Rate (SR), Fraction Class Error (FCE), Movement Delay (MD), No Classification Delay (NCD), and log-normal distribution parameters (α and β). SR is the percentage of movements and rotations which are in accordance with the user’s intentions, whether that would be a rotation, selection, left- or right-movement. For example, if SR is set to 70, it means that 70% of the imaginary movements are correct-classifications; the remaining 30% is either non-classification or wrong-classifications.

FCE (defined in Eq. (1)) depends on the SR. This parameter divides the remaining non-success imaginary movements between wrong and non-classifications. A wrong-classification is defined as the opposite of the user’s intention, and in a multi-class system there is an equal possibility between the non-intended movements. If the FCE is set to 0.8, this means that there is 80% probability that the non-success is equal to a wrong-classification as shown in Eq. (1).

\[ FCE = \frac{\sum w}{\sum w + \sum n} \]  

(1)

Here \( w \) denotes the number of wrong-classifications and \( n \) denotes the non-classifications. Non-classification represents the situation where an online BCI classifier is unable to choose either class 1 or class 2.

Two delays or latencies are implemented in the system: MD and NCD. The first one defines a constant delay between each classification, and the second one is a pause executed if a non-classification occurs. The delay represents the time from imagining the movement to an output from the classifier is obtained. Increase of these two delays should significantly decrease the communication rate. As a way to compare and measure the speed of the simulation, we use the metric Characters per Minute (CpM) defined as (Eq. (2)).

\[ CpM = \frac{C_n + 2C_d}{t} \cdot 60 \]  

(2)

Here \( C_n \) denotes the number of characters written in the final word, \( C_d \) is the number of characters deleted, and \( t \) is time elapsed in seconds. The CpM defined here is slightly different from the one defined in [14,15], where they have excluded \( C_d \). It is recommended to include \( C_d \) into the estimation of
CpM since correction is inevitable for present online BCI. By including $C_d$ we get a measurement that is an expression of the feedback system communication rate and not a measure dependent of the user’s SR.

In a Hex-O-Spell BCI setup, it is most likely that the user attempts to change from one imaginary movement, giving rotation of the arrow, to the other giving elongation and thereby performing selection as soon as the arrow points at the desired hexagon-field. The position of the arrow, when selecting a hexagon, can be modeled through a log-normal distribution of probabilities as shown in Fig. 2. The log-normal distribution in Eq. (3) is characterized by two parameters $\alpha$ and $\beta$, which becomes two additional input parameters to the BCI simulator.

$$f(x) = \frac{1}{\sqrt{2\pi} \beta} x^{-\alpha} \exp\left(-\frac{(\ln(x) - \alpha)^2}{2\beta^2}\right)$$

\textbf{2.2. Training systems}

For the simulation of the 3 training systems: Bars, Basket and Cross [19], the input parameters, SR, FCE, MD and NCD, are the same as the ones used for the Hex-O-Spell simulator. In addition, two new input parameters have to be specified: the number of trials and the step size. The training systems were used to facilitate the comparison between simulator and online results, where it was preferable to be able to run several simulations continuously without a pause. During the online BCI runs, user fatigue plays an important role in the obtained results. An increase in step size reduces the number of classifications needed to make the final selection, thus reducing the probability of fatigue. Furthermore it is important to emphasize that an increase in wrong classifications for the three training systems will prolong the session, opposite to the Hex-O-Spell program, because a classification in the training systems corresponds to an opposite movement. This means that a new classification is needed to return to status quo. The only point where the Hex-O-Spell is prolonged more than the training system is when a sequence of wrong classifications occurs, leading to a wrong selection or the arrow passes the hexagon in question. Fig. 3 visualizes the 3 training systems (a) Bars, (b) Basket and (c) Cross.

Both Bars and Basket training systems are time limited. For Bars there can be a maximum of 19 left or right imaginary movements before one of the bars are filled and a selection is

\textbf{Fig. 2 – The default probability density function used by the simulator with } $a = 1$, $\beta = 0.5$. \textbf{It illustrates the most likely position of the arrow, when a user selects the specific hexagon. The distribution can be modified to resemble online tests. In the } x\text{-axis is seen the span of one hexagon in degrees. Due to the shape of one hexagon the angles are illustrating at top as well. The } y\text{-axis is the probability of the arrow selecting at that given angle of the hexagon.}
imaginary movement should be as intended, the built-in rand-function in MATLAB has been used. Since the rand-function is pseudo-random, we propose the integration of ‘cputime’. This number is scaled and used to leap between different index points for the rand-function output, thereby making the classifier closer to true random. As long as the output from rand-function is lower than SR, a correct/intended imaginary movement will be simulated. For example, when the rand-function output is 45% and the SR is set to 70% the programmed imaginary movement will be a successful classification. The same approach is used for FCE. Here if the rand-output is below the given fraction, then a wrong classification will occur, otherwise a non-classification. Fig. 4 below clearly explains the idea behind the procedure.

3. Results

The development of a simulator for the Hex-O-Spell, and the three test systems from [19], introduces several opportunities. We obtained data from real online BCI experiments using the SMR paradigm from an associated work [19]. The data was analyzed to work with our input parameters, by statistically calculating the SR and FCE, along with the duration. The relation between our simulator and online data is illustrated in Fig. 5. The online test result (bold in red) is a randomly selected trial, visualized and compared with several trials from our simulator. These simulated trails are generated by feeding in the statistically calculated values. The figure illustrates the resulting position of the cross after a correct-, wrong- or non-classification. In this illustration, the user was asked to move the cross to the left box, which is located at the lateral position −10. Therefore, a wrong classification would illustrate a right movement, and opposite a correct classification shows a left movement. When a non-classification occurs the cross is not moved, which is illustrated with an upward line.

With a valid simulator working on among other, the Hex-O-Spell system, it is possible to improve and modify the interface
to raise the communication rate by testing different interfaces and compare the results. The Berlin Hex-O-Spell system has accomplished a communication rate as high as 7.6 CPM [12]. The experiment producing this rate was not appended with the spelled word or letter locations. To get a sense of comparability, we found input parameters that simulated this communication rate, with a Danish sentence “hej med dig” which means “Hello to you”. The input parameters are shown in Table 1. Two other sentences were also tested with length of respectively 30 and 43 characters.

One aspect that we worked on using the simulator was to expand the Hex-O-Spell from a 2-class system to a 3-class system, with an introduced dictionary hexagon. From this expansion, it was seen that the communication rate increased from 7.52 ± 0.08 to 11.17 ± 0.05 CPM. When using other input values than the one simulating the BBCI demonstration, an improvement as high as 147% was achieved.

Geng et al. [16] in their work reported that an increase in the number of classes used always follows a decrease of SR. Therefore, an investigation of the allowed drop rate of success was tested in correspondence to the increased communication rate. Here it was found that a SR drop of 55% ± 5.4 is allowed for a 3-class system to be as efficient as a 2-class system when using inputs from Table 1. Have in mind though, that these simulated results are affected by our fixed parameter values. In Fig. 6 we illustrate the effect of the “input text” parameter. Five different simulations are shown, and a dotted line representing the final CPM from the BBCI. Note the variance between trails is a result of the random nature of the simulator.

The classifier is an important aspect in a user interface, and with the simulator we investigated how the distinction between a non-classification and a wrong classification would affect the classification time in the DTU Hex-O-Spell [15]. This was done by varying the fraction class error parameter, while holding the other parameters fixed. The result is depicted in Fig. 7.

---

**Table 1 – Parameter values simulating BBCI.**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR</td>
<td>86%</td>
</tr>
<tr>
<td>MD</td>
<td>0.07 s</td>
</tr>
<tr>
<td>NCD</td>
<td>0.07 s</td>
</tr>
<tr>
<td>FCE</td>
<td>0.5 s</td>
</tr>
<tr>
<td>α</td>
<td>1</td>
</tr>
<tr>
<td>β</td>
<td>0.5</td>
</tr>
<tr>
<td>Input text</td>
<td>Hej med dig</td>
</tr>
</tbody>
</table>

---

**Fig. 5** - Number of classifications as a function of the cross position. The bold line (in red) represents cross position for one online test. The other lines represent the generated simulations from the simulator. The trials only vary with a few classifications needed compared to the online test, confirming that the simulator is valid. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

**Fig. 6** - CPM time as a function of which character is input from the Hex-O-Spell. The input letters are on the abscissa and the communication rate is displayed on the ordinate. Five trials are shown with a dotted line representing the Berlin BCI communication rate. The trials clearly visualize the fluctuation caused by letter location in the Hex-O-Spell.

**Fig. 7** - The elapsed time as a function of FCE, appended with the lower and upper boundary, and with the 25th and 75th percentile. The plot shows that an increase in FCE increases the elapsed time.
An analysis of variance yields a p-value of 0.0002. This indicates a significant difference between elapsed time for each simulation with different FCE-values.

In the test systems introduced in [19], it was observed for the Cross training system that a classification time as high as 22s in some of the trails. This long duration of where the person is constantly thinking of either a left or right hand squeezing can be exhausting. Therefore we found it interesting to use the simulator to explore the possibility to increase the step size of respectively the Cross, Basket and the Bar systems. In Table 2, the results in increasing the step size twofold for the Cross training system is provided. Observations from this table show that a reduction in steps does not affect the final simulated SR.

<table>
<thead>
<tr>
<th>SR (%)</th>
<th>Steps</th>
<th>Prior SR (%)</th>
<th>Time (s)</th>
<th>Final SR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>10</td>
<td>79.34 ± 0.10</td>
<td>05.79 ± 0.69</td>
<td>100</td>
</tr>
<tr>
<td>80</td>
<td>05</td>
<td>81.31 ± 18.26</td>
<td>03.46 ± 2.05</td>
<td>100</td>
</tr>
<tr>
<td>70</td>
<td>10</td>
<td>76.56 ± 10.84</td>
<td>06.55 ± 1.60</td>
<td>100</td>
</tr>
<tr>
<td>70</td>
<td>05</td>
<td>58.34 ± 09.63</td>
<td>05.79 ± 2.70</td>
<td>100</td>
</tr>
<tr>
<td>60</td>
<td>10</td>
<td>56.87 ± 06.15</td>
<td>11.59 ± 2.00</td>
<td>100</td>
</tr>
<tr>
<td>60</td>
<td>05</td>
<td>67.08 ± 14.93</td>
<td>04.37 ± 1.79</td>
<td>100</td>
</tr>
</tbody>
</table>

### 4. Discussion

This new simulator can be used as a tool for people interested in BCI as well as BCI researchers. Furthermore BCI interested can use this program to get acquainted with the aspects and the overall picture of a BCI system. A BCI researcher, on the other hand, could use the system to investigate how different paradigms would affect the communication rate, [18] when changing variables in the Hex-O-Spell computer interface. Therefore it differs from work such as [20] in which the performance is evaluated based on a specific system.

#### 4.1. Simulation vs. real measurements

Fig. 5 shows a close relation between the simulator and (real) online tests, with a small deviation caused by the classifier outputs minor, stochastic nature. The simulator however works with a different classification concept. A real classifier does not distinguish between success and fail. All classifications are set as success for the classifier. By implementing SR and FCE, the simulator can simulate the same response in a system with multiple classes.

#### 4.2. Testing the simulator

As described in the methodology section, an interesting way to use the simulator was to redesign the Hex-O-Spell system to work with a 3rd imaginary movement, making it a 3 class system. The expansion increases the communication rate significantly with a 147% improvement when writing longer sentences and an improvement of 48.5% was achieved with parameter values from Table 1. Though as seen in Fig. 6, this investigation is highly dependent on the input text. The result to see how low the 3-class BCI system can be compared to the 2-class BCI system yields a low SR. It is a result that can easily be misunderstood in the way that the classifier is “build” as a random generator. In some cases it would pick several wrong words, which would have decreased the classification time. Therefore, it is hard to depict anything when working with low SR values. Linear classifiers are best suited to work with a 2-class systems, so instead of expanding, one could use the simulator to use the dictionary with only 2 imaginary movements. This could be done by letting the arrow rotate with a constant speed, and then using one imaginary movement on enlarging the arrow, and the second to jump to a dictionary Hex-O-Spell. One downside of this system is that a classifier is build to classify all the time, since a control state detection is not possible.

The simulator was further used to test the effects of FCE. With a p-value of 0.0002, it was found that there is significant difference in the elapsed time in BCI systems with different FCE. This indicates that a classification system with a larger non-classification span would have a positive effect on the elapsed time.

Another improvement test was the increased step size for the training systems [19]. The final SR which is the decision that the user picked, was found to be the same even though it required fewer steps to reach it. With a communication rate that was almost twice as high it was possible to get the same results. However the increase in step size also increases the risk of a quick wrong selection as well. If the SR is not high enough, an increase in step size could have the opposite effect resulting in an increase of wrong selection. This is an example of an approach that is worth verifying on the simulator and then the next step could be the online BCI tests of the increased step size.

#### 4.3. Future development

The current form of the simulator is restricted to visualize the feedback interface of a BCI system, only simulating classified outputs and how these can be used to create an interface for the user. It is however just as informative to learn about the mechanisms in going from analog brain signals to the digital outputs. Here the simulator would expand to include parameters such as electrode placements, window length, overlap between each individual window, and a different method of classification. The method is to use the same principle as in [19], where the classifier outputs i.e. 8 values (left-, right-classification). If 5 or more is output X, then X is the final output. In this way, it is possible to model either a sensitive system or an insensitive system if we take
the former example, we instead need 7 or more to classify otherwise a non-classification will happen. The effect of this is illustrated in Fig. 7, where we saw that a system with less sensitivity, e.g. more non-classification would improve the communication rate.

The simulator works with the help of different built-in MATLAB functions. One of the functions, the “cuptime.m” gives a steadily increasing output value; the more time the simulator is running. This yields a larger processing time, and therefore affects the resulting communication rate. To avoid this factor which influences the CPM, a time independent random element should be implemented.

5. Conclusion

This study shows that a BCI simulator will be able to give a better understanding of the BCI feedback system. The simulator can be used to indicate what improvements of SR, MD or other parameters would affect the CPM and show which improvements would be best to implement. However it was found that CPM is highly dependent on the location of the letters and the sentence chosen. The findings of this BCI simulator are adaptive to a number of paradigms, as long as they all use the Hex-O-Spell system. For the DTU version of the Hex-O-Spell, a third imaginary movement was implemented in the simulator. This initial study shows that if a third imaginary movement is introduced, the communication rate was improved as high as 147%. Introduction of this third class allows the system to have 55% reduced SR and still be as efficient as a 2-class SMR system using the Hex-O-Spell program. To confirm this result, it is necessary to investigate if the simulator could simulate online Hex-O-Spell data. Currently, no data is available from the Hex-O-Spell, thus the simulator was modified to the 3 training systems: the Bars, Basket and Cross. The online results from training sessions were analyzed to inventory the correct parameters in the simulator. The simulator proved to replicate the online test-results to a satisfactory level. However there were some fluctuations due to the inherent random element.

Acknowledgments

The authors would like to thank master student Ahmad El-Madani for data exchange between the two projects. It has been a great pleasure working with him. We also appreciate the former master student Morten Jønssøn’s permission to use his “Danish Hex-O-Spell” interface layout.

References

Darius A. Rohani (M’11) received his Bachelor of Science (BSc) in Engineering (2010) from the Technical University of Denmark (DTU). In DTU he followed the BSc program in Medicine and Technology in cooperation with the University of Copenhagen, Denmark. His research interests are in Biomedical Signal Processing and Analysis. From 2010 he is following the M.Tech degree in Medicine and Technology, while working as student assistant in the department for Clinical physiologic and nuclear medicine department at Rigshospitalet, Copenhagen, Denmark. In correspondence with D.A. Rohani he is currently also working on a paper concerning heart rate variability from ECG signals (2010).

William S. Henning (M’11) received his Bachelor of Science (BSc) in Engineering (2010) from the Technical University of Denmark (DTU). At DTU he followed the BSc program in Medicine and Technology in cooperation with the University of Copenhagen, Denmark. His research interests are in Biomedical Signal Processing and Medical Imaging & Diagnostics. From 2010 he is following the Master of technology degree in Medicine and Technology, while working as student assistant in the department for Clinical physiologic and nuclear medicine department at Rigshospitalet, Copenhagen, Denmark. In correspondence with D.A. Rohani he is currently also working on a paper concerning heart rate variability from ECG signals (2010).

Carsten E. Thomsen received his M.Sc. in Electrical Engineering (1984) from Aalborg University, Denmark. In 1993 he obtained his Ph.D. degree in Biomedical Engineering from Aalborg University, Denmark. From 1984–1987 he was Research Assistant at Dept. of Medical Informatics and Image Analysis, Aalborg University, Denmark. From 1987–1992 he worked as Assistant Professor at Department of Medical Informatics and Image Analysis, Aalborg University, Denmark. In 1992 he moved to St. Bartholomew’s Hospital, London, England and worked as research fellow until 1993. Currently he is an Associate Professor at the Dept of Oral Medicine, University of Copenhagen, Denmark. His research interests are in Biomedical signal processing, intensive and critical care monitoring, epilepsy and Brain Computer Interfaces.

Troels W. Kjaer received his MD degree and subsequent Ph.D. degree from the University of Copenhagen (1992). Worked as a postdoc at National Institute of Mental Health, Bethesda, Maryland, USA 1992–1994. Worked as a senior researcher at the John F. Kennedy Institute, Glostrup, Denmark for 2 years and trained to become a specialist in clinical neurophysiology which was obtained in 2003. Has worked at department of clinical neurophysiology, Copenhagen University Hospital Rigshospitalet since 2001 – as a chief physician since 2005. Has been external associate professor at the faculty of health, University of Copenhagen since 1999. His research interests are Brain function and malfunction.

S. Puthusserypady (M’00–SM’05) received his B.Tech degree in Electrical Engineering (1986) and the M.Tech degree in Instrumentation and Control Systems Engineering (1989) from the University of Calicut, India. In 1995, he obtained his Ph.D. degree in Electrical Communication Engineering from the Indian Institute of Science, Bangalore, India. He was a Research Associate at the Department of Psychopharmacology, NIMHANS, Bangalore, India, during 1993–1996. From 1996 to 1998, he was a Postdoctoral Research Fellow at the Communications Research Laboratory, McMaster University in Hamilton, Ontario, Canada. In 1998, he joined Raytheon Systems Canada Ltd., Waterloo, Ontario, as a Senior Systems Engineer. In 2000, he moved to the National University of Singapore and worked as an Assistant Professor in the Department of Electrical and Computer Engineering until 2009. Currently, he is an Associate Professor at the Department of Electrical Engineering, Technical University of Denmark, Denmark. His research interests are in Biomedical Signal Processing, Brain-Computer Interfaces (BCIs) and Home Health Care systems.

Helge B.D. Sorensen (M’90) received his M.Sc.E.E. degree in Electrical Engineering (1985) from Institute of Electronics Systems, Aalborg University. In 1992 he obtained his Ph.D. degree in Electrical Engineering from Institute of Electronics Systems, Aalborg University. He was initially research assistant here and from 1989 to 1993, he was Assistant Professor at Institute of Electronic Engineering, Aalborg University. From 1993–1995, he was Associate Professor at Engineering Academy Denmark. Currently (since 1995) he is Associate Professor at Technical University of Denmark. He is Head of Biomedical Signal Processing research group at the Department of Electrical Engineering. His research areas are advanced biomedical signal processing and interpretation, e.g. detection of early and mature markers for neurodegenerative diseases, monitoring and detection of epileptic seizures, brain computer interfaces, and intelligent wearable biomedical devices.