# Using Time-Dependent Neural Networks for EEG Classification

Ernst Haselsteiner and Gert Pfurtscheller

***Abstract*—This paper compares two different topologies of neural networks. They are used to classify single trial electroencephalograph (EEG) data from a brain–computer interface (BCI). A short introduction to time series classification is given, and the used classifiers are described. Standard multilayer perceptrons (MLPs) are used as a standard method for classification. They are compared to finite impulse response (FIR) MLPs, which use FIR filters instead of static weights to allow temporal processing inside the classifier. A theoretical comparison of the two architectures is presented. The results of a BCI experiment with three different subjects are given and discussed. These results demonstrate the higher performance of the FIR MLP compared with the standard MLP.**

***Index Terms*—Brain–computer interface, electroencephalograph (EEG), finite impulse response multilayer perceptron (FIR MLP) networks, neural networks.**

## I. Introduction

An electroencephalograph (EEG)-based communication system, also known as brain–computer interface (BCI), can provide a new communication channel for patients with several motor disabilities, such as brain stem infarct or amyotrophic lateral sclerosis [1], [2]. EEG-based BCI can be used by "locked-in" patients to choose letters or words on a monitor by using a "language supporting program" [3].

Current BCI systems use slow cortical potentials [4] or rhythmic brain activities in the alpha and beta band to distinguish between different output classes [5]. The BCI prototype developed in Graz is based on two characteristic EEG patterns caused by two different types of motor imagination. The EEG patterns, associated with left- and right-hand movement imagination, are analyzed and classified on-line [6].

This paper addresses classification of these signals using neural networks. Neural networks are used, because they provide a well-established framework for pattern-recognition problems. The most common type, the standard multilayer perceptron (MLP), is compared to an extension of MLP, which allows temporal processing inside the classifier. This is achieved by replacing static weights with finite impulse response (FIR) filters. This type of network is therefore called FIR MLP. The motivation for using a classifier capable of temporal processing is that the patterns to be recognized are not static data but time series. Thus, the temporal information of the input data should be used to improve classification results. This can be done with a static classifier like MLP and a mapping of the temporal input data to static data. Another way is to use a classifier with temporal processing like FIR MLP. These two approaches are compared in this work.

Section II describes the Graz BCI system. The BCI experiments done in Graz are explained, and the data obtained are described. Section III deals with time series classification using neural networks. Two different topologies are introduced in Sections IV and V. In Section VI, the results of the experiments are given. This is followed by a discussion of the results in Section VII. Finally, Section VIII concludes the paper.

## II. BCI Paradigm

The subject, whose EEG is to be recorded, sits in a chair in front of a monitor and tries to control his EEG activity in accordance with some cues given on the monitor. Each trial lasts 8 s. The timing of a single trial can be seen in Fig. 1.

From second 0 to second 2, only a cross on which to focus attention at the center of the screen is given. Then, a short warning tone ("beep") is delivered, and one second later an arrow appears over the cross. This arrow points either to the right or to the left, telling the subject whether to imagine right- or left-hand movement. After 1.25 s, the arrow disappears and the feedback in form of the outline of a rectangle appears in the center of the screen. It begins to extend horizontally toward the right or left side. This horizontal bar is the only feedback given. It is controlled by a classifier based on linear discriminant analysis [7]. The length of the bar and the direction, in which it is extending (left or right), comes from that classifier. This feedback stays on the screen until the end of the trial. The subject's task is to extend the bar toward the left or right boundary of the screen, depending on the direction of the arrow presented at second 3. The sequence of the "right" and "left" trials is determined randomly.

To control the feedback, the subjects are told to imagine right- or left-hand movement depending on having an arrow pointing to the right or to the left, respectively. It is important to note that the subjects are told to remain relaxed and to avoid any actual movement while doing the imagination task.

The EEG was recorded using two bipolar leads with chlorated silver disc electrodes over left and right central areas. Channel "central left" (CL) was derived as the difference between an electrode placed 2.5 cm anterior to C3 (according to the international 10–20 system [8]), and an electrode placed 2.5 cm posterior to C3. Channel "central right" (CR) was derived similarly as the difference between two electrodes: one placed 2.5 cm posterior to C4, and the other one placed 2.5 cm anterior to C4.

Manuscript received December 24, 1998; revised April 11, 2000.

E. Haselsteiner is with the Department of Medical Informatics, Graz Technical University, Graz A-8010 Austria (e-mail: hasi@dpmi.tu-graz.ac.at).

G. Pfurtscheller is with the Ludwig Boltzmann Institute of Medical Informatics and Neuroinformatics, Graz University of Technology, Graz A-8010 Austria (e-mail: pfu@dpmi.tu-graz.ac.at).

Publisher Item Identifier S 1063-6528(00)09836-0.

This gives two channels, CL and CR, which are used in further processing. The sampling rate was 128 Hz, and signals were bandpass filtered between 0.5 and 35 Hz. This frequency band contains the alpha and the beta band. It is known [6] that the EEG patterns for BCI classification appear in these frequency bands. Because there is no perfect cutoff at 35 Hz, the sampling frequency was chosen high enough to avoid aliasing effects.

## III. Time Series Classification with NNs

The main difference between classification of static patterns and time series is the additional dimension time. In the static case, there is no relationship between different patterns. Thus, each pattern can be processed individually. If time series are to be classified, a set of *ordered* patterns has to be processed. It is the goal of time series classification to use that additional information encoded in the order of the patterns, to improve classification accuracy. A time series $x$ can be written as a sequence like

$$x = (x_0, x_1, x_2, \ldots)$$

where $x_i$ are the individual patterns of time series $x$. A trivial solution is individual classification of each $x_i$ using a standard classifier for static patterns. The classification result of the whole time series can either be the classification result of a certain element $x_i$ or some combination of the results of a set of elements of $x$.

This approach is easy to implement but has two major drawbacks. First, the information encoded in the order of the elements of $x$ is not exploited for classification, because elements are processed individually. Second, the problem of choosing an individual element $x_i$ or a set of elements to represent the whole time series is nontrivial.

That motivates a method using the temporal information for classification purpose. The elements of $x$ are fed into the classifier according to the sequence in which they appear in $x$. At every time step, the output of such a classifier should make use of the information contained in a sequence of elements. Ideally, the output depends on all elements of $x$ processed so far.

In terms of neural networks, two different ways of achieving this temporal processing are feasible. They are described in the following paragraphs.

### A. External Temporal Processing

External temporal processing means that the additional dimension time is handled outside the classifier. A subset of individual patterns $x_i$ of a time series $x$ is used to build one input pattern $p$ for the neural network. In the simplest case, the individual patterns $x_i$ are rearranged to form one larger pattern $p$. The larger pattern should ideally represent the whole time series. Thus, only the single pattern $p$ has to be processed by the neural network. For the simple case of just rearranging the individual patterns, the process is illustrated in Fig. 2.

Of course, more complicated strategies to build up pattern $p$ can be used. For example, statistic measures or fast Fourier transforms can be applied to get $p$ from $x$.

The main advantage of this approach is the use of a simple neural network for static pattern classification. Well-known architectures like MLP [9] can be applied. Also the learning process is easy, as a lot of well-known learning procedures are available. The drawback of this approach is the problem of building a pattern $p$ from the individual patterns $x_i$. There is no standard procedure, and the selection has to be done before training the network. Hence, the selection of $x_i$s is fixed and cannot be adapted during learning of the network. If simple rearrangement is used, the input dimensionality will grow linearly with the number of simple patterns $x_i$, which build up $p$. In general, this growth leads to larger networks with more weights. Thus, more training data are needed, and the generalization capability of the network is expected to decrease. Furthermore, the external temporal processing is fixed and not dependent on the actual time series $x$.

An actual implementation of this approach is discussed in more detail in Section IV.

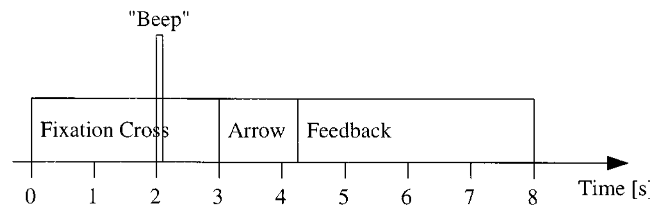


Fig. 1. Timing of a single BCI trial.

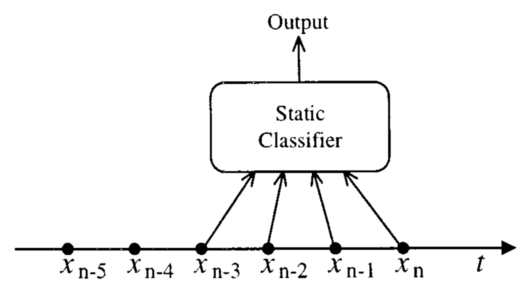


Fig. 2. Using a standard classifier for time series processing. The individual patterns $x_i$ are combined to build a new pattern, which the static classifier processes.

### B. Internal Temporal Processing

In this approach, the processing of the temporal dimension is done inside the classifier. Thus, the input to the classifier at time step $i$ is the element $x_i$ of the time series $x$. This is illustrated in Fig. 3.

The main advantage of this approach is the combination of temporal processing and classification. The processing of time is done inside the classifier and can be used directly by the classification procedure. The input dimensionality of the network does not grow with the number of time steps used. This keeps the number of free parameters of the classifier small, which makes learning easier and improves generalization capability. Besides, no fixed strategy has to be used, because the way of temporal processing can be determined by the classifier during the training phase. Given an ideal learning procedure, the strategy of temporal processing can be learned from the training examples.

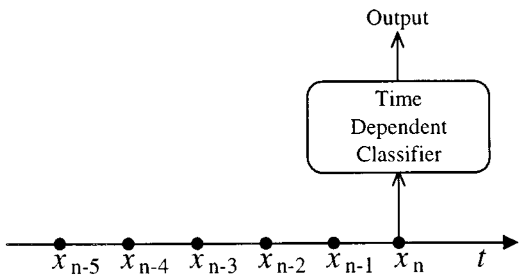


Fig. 3. At each time step, only one element of $x$ is presented to the classifier. The temporal processing is done inside the classifier.

Finding such an ideal procedure and an appropriate neural network architecture is the main problem of this approach. As these kinds of neural networks are not very common, little literature on this topic exists [10], [11].

## IV. MLP Topology

In this section, an external temporal processing approach using a standard multilayer perceptron [9] is described. The temporal processing is done outside the classifier. The given time series $x$ is transformed into one single pattern $p$, which is the input to the neural network.

This transformation is done by combining a set of elements $x_i$ of $x$ into one input vector $p$. The selection of the indexes $i$ is based on a priori knowledge about the given time series and is limited by computational resources. As there are $2^l$ possible subsets, where $l$ is the number of elements of $x$, it is impossible to check all possible subsets. Hence, the experiments have to be restricted to a few different sets. This is one of the drawbacks of this approach.

Another disadvantage of this topology is the strong dependence of the number of free parameters of the classifier (i.e., the weights of the neural network) on the size $r$ of the chosen subset of indexes $i$. Each time step is represented by a vector of some fixed dimension $d(x_i)$. Thus, for the dimension of $p$ follows

$$d(p) = r \cdot d(x_i).$$

Using an MLP for classification of $p$, the number of input units equals $d(p)$. The total number of weights $W_{\mathrm{MLP}}$ of an MLP with two fully connected layers, two output units, and $m$ units in the hidden layer can be calculated as

$$W_{\mathrm{MLP}} = r \cdot d(x_i) \cdot m + 2 \cdot m + m + 2. \tag{1}$$

The two last terms are caused by the bias weights of each neuron. Fig. 4 illustrates the described topology. Two output units were used, because the two classes were encoded as (0, 1) and (1, 0), respectively. In this two-class problem, the use of only one output unit is also possible and yields the same results.

In the special case of $r = 1$, no temporal processing is done and the result of the classification depends only on one single time step.

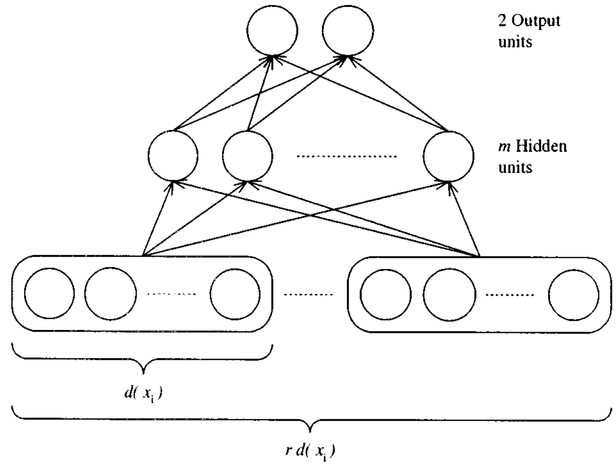


Fig. 4. Standard MLP topology with $r \cdot d(x_i)$ input units, $m$ hidden units, and two output units.

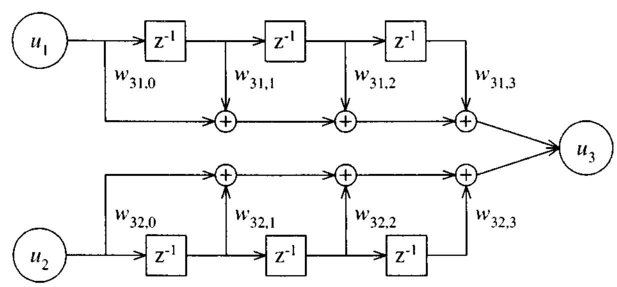


Fig. 5. The units $u_1$ and $u_2$ are connected to unit $u_3$ using FIR filters instead of two scalar weights. Each filter is of the third order and has four coefficients or weights. Nonlinearities can only appear inside units $u$.

## V. FIR MLP Topology

The FIR MLP topology [10], [11] is a natural extension of the standard MLP. The temporal processing is done using FIR filters [12]. The static weights of a standard perceptron are replaced by such filters. The number of delay units in each filter is the order $q$ of the filter. Fig. 5 shows a network with three units connected by two third-order FIR filters.

A filter of order $q$ uses $q+1$ different time steps to calculate the output of the filter, which is the input for the next neuron. Thus, the filter needs $q+1$ coefficients, which are often called weights in the context of neural networks. By using these filters, the temporal processing takes place inside the classifier, and the strategy used is encoded in the weights of the filters. Thus, this strategy can be learned from given examples by adapting the weights.

Fig. 6 displays the structure of such a classifier.

In this study, all connections between neurons were replaced by FIR filters. The order of the filters was fixed to $q_{\mathrm{hid}}$ for the hidden layer and to $q_{\mathrm{out}}$ for the output layer. Of course, each filter can be of different order. However, it is not clear how to choose the order for each filter in advance. For this reason, the order of the filters in each layer was set to the same value. During learning, the order of individual filters can be decreased by setting corresponding coefficients to zero or close to zero.

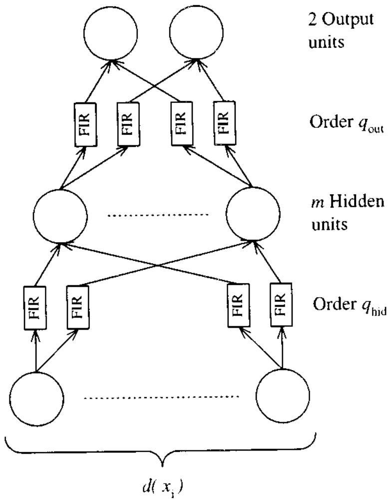


Fig. 6. The static weights are replaced by FIR filters. The temporal processing is done inside the classifier.

If the order of a filter is set to zero, this filter will become a static weight. If the order of all filters of the network is set to zero, the FIR MLP will default to a standard MLP.

It is also possible to include an additional gain term in each filter, which multiplies the output of the filter by some scalar. As described in [13], this gain term should speed up and improve learning. If the output of a filter needs to be amplified, it will be easier to adjust the single gain term than to adjust each individual coefficient. However, the gain term is also an additional free parameter, which makes learning harder. Experiments with and without gain terms showed that for the Graz BCI application networks without gain terms performed better.

The FIR MLP architecture can also be considered a kind of time-delay neural network (TDNN), proposed by Waibel [14]. It can be shown that both architectures are functionally equivalent.

An advantage of the FIR MLP topology is the reduction of free parameters compared to the MLP with external temporal processing. The number of free parameters $W_{\text{FIR}}$ of a network like the one shown in Fig. 6, with $m$ hidden units and two output units, is

$$W_{\text{FIR}} = (q_{\text{hid}} + 1) \cdot d(x_i) \cdot m + 2 \cdot m \cdot (q_{\text{out}} + 1) + m + 2. \tag{2}$$

The term $d(x_i)$ is the dimension of an element of the time series. The two last terms are caused by the bias weights of each neuron.

The following example is used to demonstrate the reduction of free parameters by using FIR MLP instead of MLP. The maximum number of elements of the time series that can be used for classification is called memory depth $n_{\max}$. For the MLP classifier with external temporal processing, $n_{\max}$ equals the number of elements $r$ used to form the input vector $p$ (see Fig. 4). In the case of the FIR MLP architecture shown in Fig. 6, the memory depth equals the number of delays between the input and the output of the network plus one, because the current input is also used. This yields the following equation for $n_{\max}$:

$$n_{\max} = q_{\text{hid}} + q_{\text{out}} + 1. \tag{3}$$

To compare an MLP to an FIR MLP topology, $n_{\max}$ is fixed to eight for both architectures. The dimension of a single element of the time series is set to 12, because this is the value that was used in the experiments (see Section VI). Based on this assumptions, the number of free parameters is calculated. This gives the number of weights necessary to get a memory depth of 8. Using (1), this gives $W_{\text{MLP}}$

$$W_{\text{MLP}} = 8 \cdot 12 \cdot 4 + 2 \cdot 4 + 4 + 2 = 398.$$

Using (2) requires one to set the filter order for the hidden and the output layer. To satisfy the assumption, $n_{\max} = 8, q_{\text{hid}} = 4$, and $q_{\text{out}} = 3$ were chosen. This gives $n_{\max} = 8$ according to (3). From (2) follows

$$W_{\text{FIR}} = (4 + 1) \cdot 12 \cdot 4 + 2 \cdot 4 \cdot (3 + 1) + 4 + 2 = 278.$$

This shows the significant reduction of free parameters caused by the architecture of the FIR MLP network providing the same memory depth $n_{\max}$ as an MLP network.

Another important difference between FIR MLP and MLP is the more efficient use of the training data by the FIR MLP network. There are two basic procedures for training FIR MLP networks: the so-called temporal backpropagation by Wan [10], [15] and the algorithm by Back and Tsoi [11] Both algorithms produce weight updates for each element of the time series. They differ slightly in the way of calculating local errors during backpropagation [13].

In the case of MLP, there is only one weight update per time series. Thus, for a time series consisting of $l$ elements, there are $l$ times more weight updates in the FIR MLP architecture than in the MLP network.

The reduction of free parameters and the more efficient use of the training data are two advantages of the FIR MLP, which make the estimation of the weights easier and able to yield better generalization.

However, there is also a drawback of the FIR MLP approach. Target values have to be provided for each element of the time series, instead of having one target for the whole time series. To overcome this problem, the same target can be used for all time steps, or more advanced techniques like adaptive dynamic targets [16] can be used. It is also possible to provide targets for certain time steps only and generate no errors at the remaining time steps [17].

## VI. RESULTS

The experiments were carried out using data of three different subjects. The subjects are referred to as f7, g3, and i2 in the following text. All the data were obtained from the BCI experiment described in Section II. The trials were manually inspected for artifacts, and only artifact-free trials were used. Table I shows how many artifact-free trials of each subject were used and which period of time was used for each subjects. The selection of individual periods for different subjects is based on results of previous studies [6]. The use of different periods for each subject reflects the different timing of characteristic EEG patterns caused by imagination. All the moments of time given refer to the BCI experiment, as described in Fig. 1.

The raw EEG data was not classified directly. An adaptive autoregressive (AAR) model of order six was used to characterize the EEG. AAR models are a general tool for signal modeling. Such a model was continuously adapted to the current EEG signal, and the coefficients of the AAR model were used as features for the classifier. The model order of six was chosen based on previous studies on describing EEG data with AAR models [18]. Because the coefficients of the AAR model change much more slowly than the raw EEG, which they describe, only each sixteenth time step was used. This gives a temporal resolution of 8 Hz. Each of the two EEG channels, CL and CR, was characterized by a sixth-order AAR model. For classification, all coefficients of both models were used, yielding 12-dimensional feature vectors.

The MLP architectures were trained with a standard backpropagation procedure as described in [9]. The training was stopped when the error rate on a validation set stopped to decrease or started to increase. The FIR MLP architectures were trained using Wan's temporal backpropagation. This algorithm showed slightly better performance than the algorithm of Back and Tsoi. The coefficient of the AAR models, which are used as features, change slowly. That might be the reason for the better performance of Wan's algorithm. This agrees with the results presented in [13]. The stopping criterion of the FIR MLP training was also the error rate on a validation set.

All results shown were obtained using five-fold cross validation. Therefore, the data set of each subject was divided into three sets. The training set (60% of the data) was used to train the networks, the validation set (20% of the data) was used for the stopping criterion, and the test set (20% of the data) was used to calculate the final classification accuracy. After splitting up the data, five classifiers with different initial conditions were trained. They were used to form a committee by averaging their outputs [19]. This procedure of splitting up the data set, doing the training, and building committees was done five times for each experiment on each subject. The average results on the test sets and 95% confidence intervals are given for each experiment.

### A. MLP Results

In the first experiment, only one time step was used to classify the time series. This leads to an architecture shown in Fig. 4 with $r = 1$. The number of hidden units $m$ was varied between one and four. The time step $t$ used for classification was varied within the useful range of each subject (see Table I). The two best results of each subject are given in Table II.

The table also gives the moment of time, which yielded the minimal error rate. It can be seen that the error rate achieved strongly depends on the subject. The three subjects chosen for this study represent a wide range. Subject g3 performs very well, subject f7 is kind of average, and subject i2 gives only poor results.

Furthermore, architectures using two time steps for classification were used $(r = 2)$. The distance $\tau$ between the two time steps was fixed. This means the input vector to the classifier was a combination of the feature vectors at time step $t$ and time step $t + \tau$. The value of $t$ was varied within the useful range of each person. $\tau$ was varied between 0.125 and 0.5 s with a step size of 0.125 s. The number of hidden units $m$ was varied between one and four. Table III shows the two best results of each subject.

TABLE I
NUMBERS OF ARTIFACT-FREE TRIALS AND PERIOD OF TIME USED FOR CLASSIFICATION FOR EACH SUBJECT

| Subject | Left Trials | Right Trials | Period used [s] |
|---|---|---|---|
| f7 | 395 | 393 | 3.50 - 5.00 |
| g3 | 315 | 316 | 3.00 - 7.00 |
| i2 | 290 | 292 | 3.50 - 6.75 |

TABLE II
ERROR RATES USING MLP ARCHITECTURE WITH ONE TIME STEP AND $m$ HIDDEN UNITS

| Subject | $m$ | Error Rate [%] | $t$ [s] |
|---|---|---|---|
| f7 | 3 | 15.8 ± 2.4 | 4.375 |
| f7 | 1 | 16.2 ± 2.2 | 4.375 |
| g3 | 1 | 5.7 ± 1.1 | 6.500 |
| g3 | 4 | 5.9 ± 0.8 | 6.500 |
| i2 | 1 | 24.0 ± 3.9 | 6.250 |
| i2 | 4 | 24.1 ± 2.2 | 5.750 |

TABLE III
BEST ACHIEVED ERROR RATES USING MLP ARCHITECTURE WITH TWO TIME STEPS AND PARAMETERS $m$ AND $\tau$

| Subject | $m$ | $\tau$ [s] | Error Rate [%] | $t$ [s] |
|---|---|---|---|---|
| f7 | 3 | 0.250 | 14.1 ± 3.2 | 4.375 |
| f7 | 2 | 0.250 | 14.4 ± 1.8 | 4.375 |
| g3 | 2 | 0.500 | 4.9 ± 0.8 | 6.500 |
| g3 | 4 | 0.500 | 5.2 ± 1.0 | 6.500 |
| i2 | 3 | 0.500 | 22.8 ± 2.8 | 5.000 |
| i2 | 2 | 0.500 | 22.8 ± 3.0 | 5.000 |

Experiments using three different time steps were done, but within the used combinations of three time steps no significant improvements were found. Furthermore, the number of possible subsets becomes too large to do systematic tests. That is why no further investigations on sets with three or more time steps were done.

### B. FIR MLP Results

The results were obtained with an FIR MLP architecture, as described in Section V. In this work, no gain terms were used. The orders of the filters of the first and the second layer were fixed to $q_{\text{hid}}$ and $q_{\text{out}}$, respectively. Different values for these two parameters as well as for the number of hidden units $m$ were used. The number of hidden units $m$ was varied between one and four. A total of ten different combinations of $m, q_{\text{hid}}$, and $q_{\text{out}}$ were investigated. For each combination and each subject, cross validation, as described previously, was applied. The two best results of each subject are given in Table IV.

## VII. DISCUSSION

### A. Using Multiple Time Steps

From the results in Tables II and III, it can be concluded that using multiple time steps to classify a given time series does

TABLE IV
ERROR RATES USING FIR MLP ARCHITECTURE WITH $m$ HIDDEN UNITS, $q_{\mathrm{hid}}$ DELAYS IN THE FIRST LAYER, AND $q_{\mathrm{out}}$ DELAYS IN THE SECOND LAYER

| Subject | $m$ | $q_{hid}$ | $q_{out}$ | Error Rate [%] | $t$ [s] |
|---|---|---|---|---|---|
| f7 | 2 | 4 | 2 | 12.0 ± 0.5 | 4.500 |
| f7 | 2 | 1 | 0 | 12.0 ± 0.6 | 4.375 |
| g3 | 2 | 1 | 0 | 4.3 ± 0.3 | 6.000 |
| g3 | 2 | 1 | 2 | 4.3 ± 0.7 | 6.000 |
| i2 | 2 | 1 | 0 | 21.6 ± 0.8 | 5.750 |
| i2 | 2 | 1 | 2 | 21.6 ± 0.8 | 5.625 |

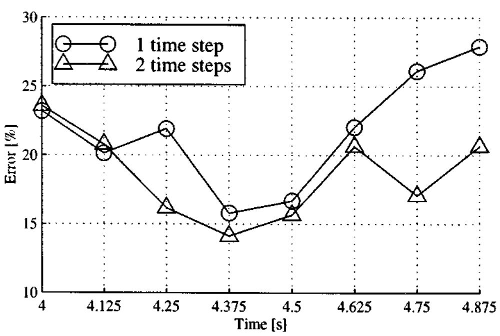


Fig. 7. Using one time step (circular marks) versus two time steps (triangular marks) for classification of subject f7.

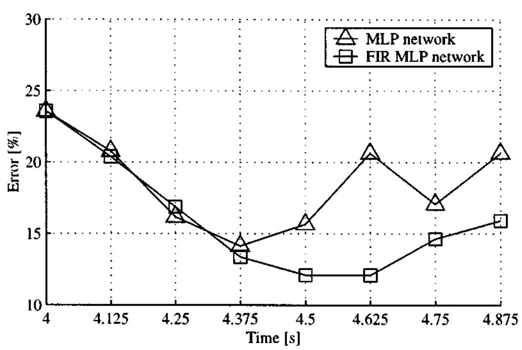


Fig. 8. Using MLP network (triangular marks) versus FIR architecture (square marks) for classification.

reduce the error rate. Fig. 7 shows the performance of the best classifier using one time step versus the performance of the best classifier using two time steps for classification of subject f7. The error rate is displayed as a function of time $t$, where $t$ refers to the moment of time of a BCI trial, as shown in Fig. 1. Only the values at the marked positions were calculated. They were connected by lines for illustration purposes.

Using a $t$-test with a level of significance $\alpha = 0.05$, it can be shown that the results of subject f7 for $t = 4.375$ s are significantly better if two time steps are used for classification.

Furthermore, the results of the other two subjects, g3 and i2, improved as well when two time steps instead of one were used. This presents strong evidence that using temporal information yields better classification accuracy.

### B. MLP Versus FIR MLP

The results given in Tables III and IV demonstrate the improvement caused by using FIR MLP networks instead of MLP networks. Fig. 8 illustrates this by showing the best result achieved using MLP architecture and the best result using FIR MLP networks of subject f7.

With subject f7, the best error rate achieved of 12.0% using FIR MLP is significantly better than the best error rate of 14.1% achieved by the MLP approach. This can be shown using a $t$-test with a level of significance $\alpha = 0.05$. Also, the results of the FIR MLP network of the other two subjects, g3 and i2, are significantly better than those of MLP architectures.

The use of FIR MLP architectures also reduces the variations between different classifiers trained on the same data. This can be seen by comparing the sizes of the 95% confidence intervals given in Tables III and IV.

There are two main reasons for the better performance of the FIR MLP architecture. On the one hand, the classifier is much better suited for exploiting temporal information contained in the time series to be classified. On the other hand, the training of this classifier is much more efficient because all time steps of the time series can be used, which increases the number of training samples and helps to produce a more robust classifier.

## VIII. CONCLUSION

In this paper, a comparison between using standard MLP classifiers and using FIR MLP networks for single trial EEG classification is given. In the case of MLP networks, it is shown that the error rate can be reduced if the classifier uses a feature vector comprising two time steps of the given time series instead of one single time step.

The best results of the MLP architecture are compared against those of an FIR MLP network on the same problem. The achieved error rates as well as the robustness of the FIR MLP classifiers demonstrate the better performance of the FIR MLP networks in comparison with the standard MLP networks.

## REFERENCES

[1] J. Vidal, “Toward direct brain-computer communication,” *Annu. Rev. Biophys. Bioeng.*, pp. 157–180, 1973.
[2] J. Wolpaw and D. McFarland, “Multichannel eeg-based brain-computer communication,” *Electroenceph. Clin. Neurophysiol.*, vol. 90, pp. 444–449, 1994.
[3] J. Perelmouter, B. Kotchoubey, A. Kbler, E. Taub, and N. Birbaumer, “A language supporting program for thought translation devices,” *Automedica*, 1998, to be published.
[4] B. Kotchoubey, H. Schleichert, W. Lutzenberger, and N. Birbaumer, “A new method for self-regulation of slow cortical potentials in a timed paradigm,” *Appl. Psychophysiol. Biofeedback*, vol. 22, no. 2, pp. 77–93, 1997.
[5] 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. Eng.*, vol. 6, no. 3, pp. 316–325, 1998.
[6] G. Pfurtscheller, C. Neuper, D. Flotzinger, and M. Pregenzer, “EEG-based discrimination between imagination of right and left hand movement,” *Electroenceph. Clin. Neurophys.*, vol. 103, no. 6, pp. 642–651, 1997.
[7] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York: Academic , 1990.
[8] E. Niedermeyer and F. L. da Silva, *Electroencephalography*, 4th ed. New York: Lippincott Williams & Wilkins, 1998.

[9] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing*. Cambridge, MA: MIT Press, 1986, vol. 1.
[10] E. A. Wan, "Time series prediction by using a connectionist network with internal delay lines," in *Time Series Prediction: Forecasting the Future and Understanding the Past*, A. S. Weigend and N. A. Gershenfeld, Eds. Reading, MA: Addison-Wesley, 1993, pp. 195–217.
[11] A. D. Back and A. C. Tsoi, "FIR and IIR synapses, a new neural network architecture for time series modeling," *Neural Computat.*, vol. 3, pp. 375–385, 1991.
[12] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1998.
[13] A. D. Back and A. C. Tsoi, "Aspects of adaptive learning algorithms for FIR feedforward networks," in *Proc. 1996 Int. Conf. Neural Information Processing*, vol. 2, 1996, pp. 1311–1316.
[14] A. Waibel, "Modular construction of time-delay neural networks for speech recognition," *Neural Computat.*, vol. 1, pp. 39–46, 1989.
[15] S. Haykin, *Neural Networks*. New York: Macmillan, 1994.
[16] E. Haselsteiner, "Time series classification using adaptive dynamic targets," in *Proc. IEEE Workshop Neural Networks for Signal Processing NNSP'99*, Madison, WI, 1999, pp. 243–252.
[17] M. I. Jordan, "Serial order: A parallel distributed processing approach," Institute for Cognitive Science, Univ. of California, Tech. Rep., 1986.
[18] A. Schlögl, D. Flotzinger, and G. Pfurtscheller, "Adaptive autoregressive modeling used for single-trial EEG classification," *Biomedizinische Technik*, vol. 42, no. 6, pp. 162–167, 1997.
[19] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford, U.K.: Clarendon , 1996.

**Ernst Haselsteiner** received the M.Sc. degree in telematik from Technical University Graz, Austria, in 1997, where he is now pursuing the Ph.D. degree at the Institute of Biomedical Engineering, Department of Medical Informatics.

He also doing some projects with Siemens AG, Austria, in the area of pattern recognition. His research interests include machine learning, neural networks, image and signal processing, and pattern recognition.

**Gert Pfurtscheller** received the M.Sc. and Ph.D. degrees in electrical engineering from Graz University of Technology, Austria, in 1964 and 1966, respectively.

He is a Professor of medical informatics at the Institute of Biomedical Engineering, Graz University of Technology. Since 1982, he has been Head of the Department of Medical Informatics, and since 1987 Director of the Ludwig Boltzmann-Institute for Medical Informatics and Neuroinformatics. He has authored more than 300 scientific articles and four books. His scientific interests and current research work include biomedical signal processing with special emphasis on electrical brain activity analysis, brain–computer communication systems, and medical expert systems.