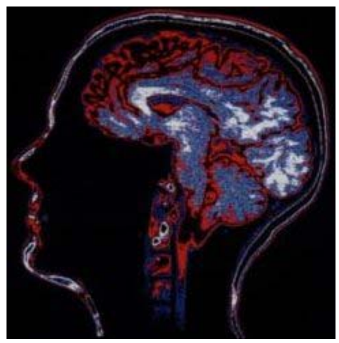
Brain states analysis for direct brain-computer communication



Julien Freudiger

Brain states analysis for direct brain computer analysis

Student Julien Freudiger
Prof. Dr. Touradj Ebrahimi
Responsable assistant Gary N. Garcia

Summer 2003

Index:

Subject:

A direct brain-computer communication device that allows a user to directly interact with a computer rendered environment through his thoughts is called a brain-computer interface (BCI). During the last year, the Signal Processing Laboratory of the Swiss Federal Institute of Technology developed a BCI based on the analysis of electroencephalogram measurements (EEG) [1] [2] [3].

The EEG signals are analysed and mapped into actions inside the computer rendered environment. According to neuroscience results, the spectral properties of EEG exhibit noticeable changes when the user performs given mental activities.

Besides the spectral changes, the temporal properties of EEG signals can also be monitored. The evolution of these properties gives complementary information about mental activities patterns and their transitions.

The purpose of this paper is to analyze the EEG time series evolution using typical microstate segmentation (MSS) [4] implemented in Matlab.

Abstract:

In order to permit a brain computer efficient communication, it is important to dispose of an efficient algorithm to decode the brain electrical activity. We will focus our attention on an algorithm based on microstates segmentation of the brain electrical activity. First of all, we are going to use electroencephalogram measurements (EEG, 10/20 international system) to collect the electrical activity of the brain. Each sixteen electrodes of the EEG will sample the data at the same rate, forming a matrix of measures.

The algorithm is based on the hypothesis that a particular mental activity will generate momentary detectable potential scalp maps (Event Related Potentials, ERP). A mental activity can therefore be seen as a sequential organization of scalp maps, called microstates.

To find the best microstates representing a particular mental activity is an exiting challenge. We are going to express the sets of data obtained through EEG as time series of nonoverlapping microstates with different intensities. The algorithm will then converge to a set of microstates representing the data with minimum error.

Another inherent problem is that of the ideal number of microstate. How many scalp maps should we use to represent our set of data? The cross validation method is the more adequate for resolving this kind of problems. Applying this method to the data will give us an approximate number of states with a certain error.

Improvements of the algorithm are also introduced. There were necessary to guarantee better results and more efficiency. Results are presented in the Comparison section, in which the qualities of the algorithm are underlined by some results analysis. We also compute its complexity and have created an indicator to improve our utilisation of the algorithm. The appendix provides the mathematical demonstrations of the algorithm formulas and focuses its attention on trying to explain all the difficult to understand concepts.

1. Introduction:

1.1. What is a Brain-Computer Interface (BCI)?

The basic idea is to develop another communication way between a human and a machine. Since the creation of computer rendered environments, the computers power has been increasing to an amazing speed. Each year, the possibilities offered by a computer are growing extraordinarily. However, the communications with a machine have unfortunately not significantly evolved. We still have to content ourselves of an old keyboard, a mouse and a screen display. It is undeniable that it would be a wonderful evolution to improve our interactions with the machines. Following this idea, we are concerned by developing a new way of communication using the brain.

Based on measures of the electrical activity of the brain, our goal is to discover the user's thoughts in order to execute his orders. To accomplish this prowess, we use the natural properties of the brain: to identify a particular mental activity of the subject, we observe the variation of his mental activities in frequency and in time using an appropriate material (electroencephalogram, EEG, see 1.3.). A mathematical algorithm will next analyse the collected data and guess about the subject thoughts.

This technology offers creating a completely new way of communication offering lots of possibilities. For example, for persons with movements' disabilities, a BCI could help them to command their electric wheelchair without needing somebody else help, giving them more independence.

As another example, we can image substitute some damaged nerves by computers intercepting the brain messages and redirecting them to the muscle. For people having to deal with epilepsy crisis, it would be interesting to improve their knowledge of the different processes that stimulate a crisis and try to avoid them, controlling their own brain as another muscle.

1.2. How does the brain works?

Our brain is mostly composed of neurons interconnected to each others forming an enormous network (Figure 1). They communicate together through their axons using small electric impulses (μV).

Figure 1: neurons connections.

During a particular mental activity, we can observe electric potential variations of the brain active regions. Neurons are generating small electrical variation that summed over a region give a potential variation in the space. These variations can be decomposed in series of electrical maps. This means that a mental activity can be seen as a sequential continuation of brain electrical states.

Based on recordings of the brain electrical activity, we would like to reflect as well as possible the dynamic of the functional states of the brain and identify the states representing a mental task as well has possible in order to be able to recognize later on the same mental activity.

1.3. How can we make it possible?

In order to measure the electrical activity of the brain in a non-invasive way, nowadays, the best choice is to employ electroencephalogram (EEG) material [1]. EEG measurements have proved their utility in the medical world and are not expensive. The 10/20 international system is used for positioning the electrodes on the scalp. This norm defines 16 electrodes positions best positioned in order to represent as well as possible the electrical activity of the brain (Figure 2).

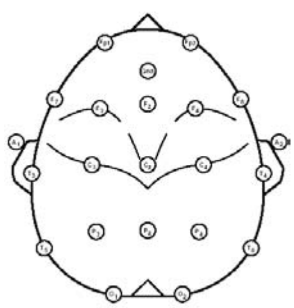


Figure 2: 10/20 system placement.

Mental activity can be observed as potential variation measured by the electrodes (Figure 3). The scalp electromagnetic field reflects the source distribution in the brain.

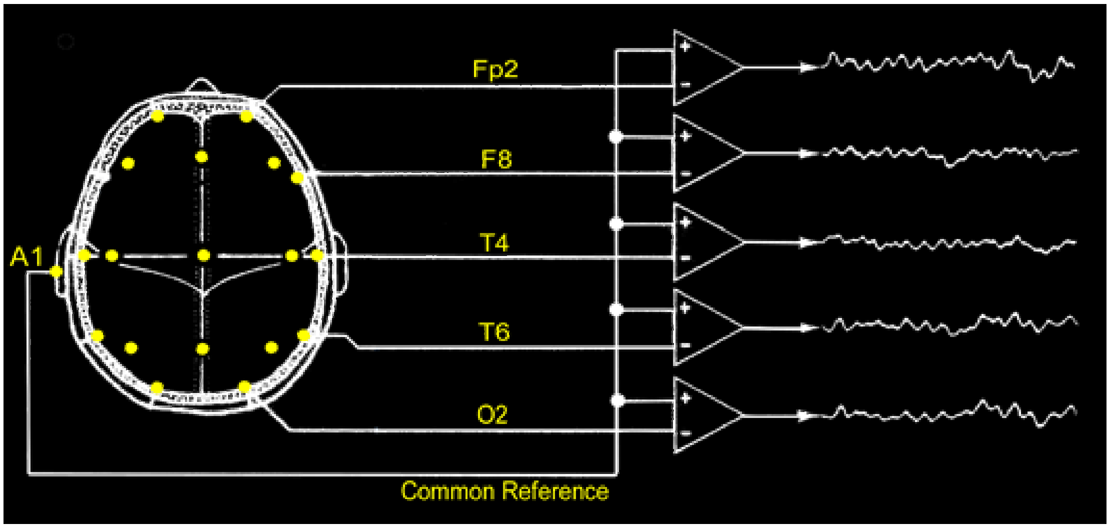


Figure 3: mental activity recordings.

Why not using another more sophisticated way to measure the brain electrical activity? We could for example use the Magnetic Image Resonance method (IRM)? Let's remember our first objective. All we would like to do is to identify some pattern in the mental activity of the subject in order to give him an immediate feed back. The IRM are useful to describe the brain form and to study the blood fluctuations but they don't measure the electrical activity of the patient. It is easy to imagine that there is a correlation between the electrical activity and the blood fluctuations in the brain, but the latency (variation in time) is fundamentally different!

Another more interesting idea is to use microchips introduced directly to the contact of the brain. It poses some new problems like difficulties of avoiding infections in the human body, or transmitting the data measured at the contact of the brain to the computer (through wireless communication?). Moreover, this kind of solutions will just measure the brain activity in a very close region of the brain. Studying the ecological states of a few trees randomly selected in a forest, is not enough to conclude about the ecological state of the entire forest. So, it would be necessary to increase the number of introduced microchips augmenting at the same time the discomfort of the user.

Moreover, integrating chips in the human corps is not yet very well defined in the medicine world and neither in the ethic of the humanity. In conclusion, the EEG systems based on 10/20 international positioning, is efficient enough for our objective: detecting some special mental activities.

number of samples t

	-3.3001	0.4227	-0.6307	-3.0274	-3.6069		-0.5506	0.5099	-0.0487
	0.5917	-0.4685	-2.7475	-3.5181	-1.9794		1.7459	5.8961	8.4626
	-6.0412	-5.3666	-8.3811	-9.8704	-7.4653		-2.0519	-0.2716	-1.4554
	-6.1430	-4.4444	-6.8840	-8.2142	-4.1644		1.2228	3.2313	2.5710
	-1.1795	-0.4563	-2.8554	-3.9883	0.5829		5.4314	7.7453	7.3331
	-4.1720	-5.3332	-8.5824	-10.7859	-8.0743		-4.3975	-1.5014	-0.9399
	-1.0386	-1.2687	-4.0254	-5.0138	-3.7450	...	-2.6659	-3.0663	-3.5344
electrodes	-8.7889	-7.1862	-8.6636	-7.3227	-3.9211		0.3156	1.7333	1.5516
	-1.0109	-1.4033	-2.3331	-0.6622	1.2799		3.8749	5.4914	5.6513
	0.4586	0.3674	-0.4808	-0.2714	1.4573		3.9659	6.2520	6.5310
	-2.0777	0.1491	1.1471	0.5548	1.0192		0.4386	-1.1359	-5.5009
	-2.1974	-0.0177	0.3960	-0.8391	-0.0322		1.6317	1.5686	-1.7345
	1.3035	2.4808	1.7668	-0.2381	-1.0056		0.4330	-0.2145	-3.2802
	-1.5028	0.8635	-0.0064	-0.0500	0.7408		2.7475	1.0460	-2.0664
	-1.2996	1.5810	-0.7814	-3.5349	-4.0353		-3.2323	-5.1097	-7.3554
	-2.2864	0.8430	-0.9535	-3.5383	-3.0781		-4.0847	-7.6961	-12.2004

Figure 4: matrix of data.

To record the electrical activity, we sample the mental activity using 16 electrodes and store them in a matrix (16xNumber of measurements, Figure 4). Each column contains the 16 values collected in the electrodes at a certain time t. For the analysis of the values, Matlab will be employed, having an important strength in mathematical analysis.

2. Microstate segmentation (MSS):

2.1. Principles:

The brain electrical activity recordings consist of measurements of the scalp electric potential field. We focus our attention on electrical activity performed for particular tasks called Event Related Potentials (ERP).

Instead of viewing the multichannel records of event related potential data as wave shapes, we will analyze them as sequences of momentary electric potential distributions maps (Figure 5).

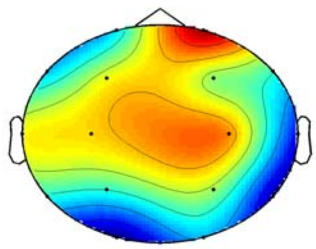


Figure 5: scalp electric potential image where the red regions are the positive ones and the blue regions the negative ones.

If we observe the evolution of the scalp map configuration, it will noticeable that configurations are stable during small time segments. According to neuroscience's hypothesis [4], these stable map configurations reflect functional microstates of the brain. It means that these momentary spatial distributions represent modes of information processing. A mental activity can therefore be seen as a sequence of nonoverlapping states with variable duration and intensity.

An interesting way to represent the given set of data is to determine some normalized vectors T representing the different microstates observed. By the way, it is easy to label the set of data: each time t will be considered as owning a microstate with a particular intensity stored in a matrix A; mathematically:

$$\vec{V}_t = \sum_{k=1}^{N_\mu} a_{kt} \cdot \vec{T}_k \tag{1}$$

With constraints guaranteeing nonoverlapping states:

$$\begin{aligned} & a_{k_1 t} \cdot a_{k_2 t} = 0, \forall k_1 \neq k_2, \forall t \\ & \sum_{k=1}^{N\mu} a_{kt}^2 \geq 0, \forall t \end{aligned} \tag{2}$$

Where N_μ is the number of different states, $\vec{V}_t$ a vector of scalp electric potential measurements at time instant t, T_k representing the k-th microstate vector and a_{kt} the k-th microstate intensity at time instant t. We are searching to represent our measurements matrix V with nonoverlapping microstates. So, at each time instant $t \in \{1,...,N_t\}$, all a_{kt} must be null except for one. Formally, it is represented by constraints (2).

Example:

If we have $N_\mu = 4$:

$$T = \begin{bmatrix} \vec{T}_1 & \vec{T}_2 & \vec{T}_3 & \vec{T}_4 \end{bmatrix} \qquad and \qquad A = \begin{bmatrix} 0 & a_{12} & 0 & 0 & \\ a_{21} & 0 & 0 & 0 & ... \\ 0 & 0 & 0 & a_{3t} & \\ 0 & 0 & a_{43} & 0 & \end{bmatrix};$$

An interesting point of view to get the idea is to imagine that each microstate represent a letter of the alphabet. The series of states form words and phrases that are the voice of the brain. These microstates are the way we use to listen to the brain. So on, we are not interested in just one microstate but rather in series of microstates. They express the mental activity the user is performing.

Estimating the number of microstates (using cross validation) and finding the best microstates (MicroState Algorithm, MSS) are two different problems inherent to the method.

2.2. Estimating the microstates:

2.2.1. The formula:

The MSS algorithm has two necessary initial conditions:
First, it needs a vector of data for which he will find the best microstates (V). Second, it needs some initials microstates, randomly selected from the set of data and normalized (T). Then by iteration, it will find the parameters a and T that minimize the following function:

$$F = \frac{1}{N_t \cdot (N_s - 1)} \cdot \sum_{t=1}^{N_t} \left\| V_t - \sum_{k=1}^{N_\mu} a_{kt} \cdot T_k \right\|^2 \tag{3}$$

2.2.2. Geometrical representation and labeling:

A geometrical representation of the microstate problem can help us to illustrate the algorithm. Consider a 2D plane having as the x axis the potential of one electrode and as the y axis the potential of another electrode (it is equivalent to represent the problem with just two electrodes rather than 16, and the illustration is much easier).

On this plane, a microstate is characterized by a point located at unit distance from the origin (because of its normalization). All points lying on the line going from the origin toward the microstate belong to the same microstate. The intensity of the microstate representing a measured point is related to the distance from the origin to the point (Figure 6). In the following example, we have five microstates m1 to m5.

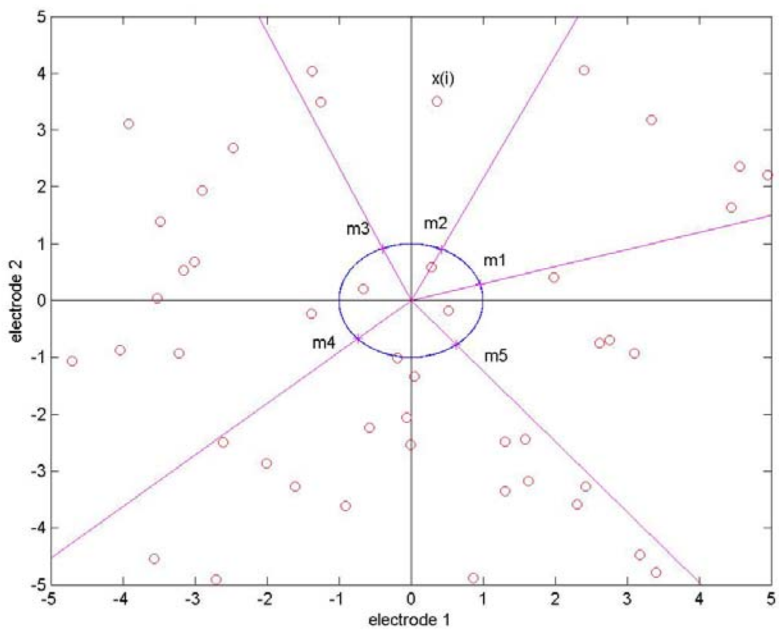


Figure 6: Geometrical representation of the microstate model.

We have to label each circle points representing measures at a certain time. The MSS algorithm calculates the Euclidian distance between a point and the microstates, and chooses the shortest one (Figure 7). In our example, clearly d1 < d2, it implies that the point x(i) will be labeled as belonging to microstate 2 (m2).

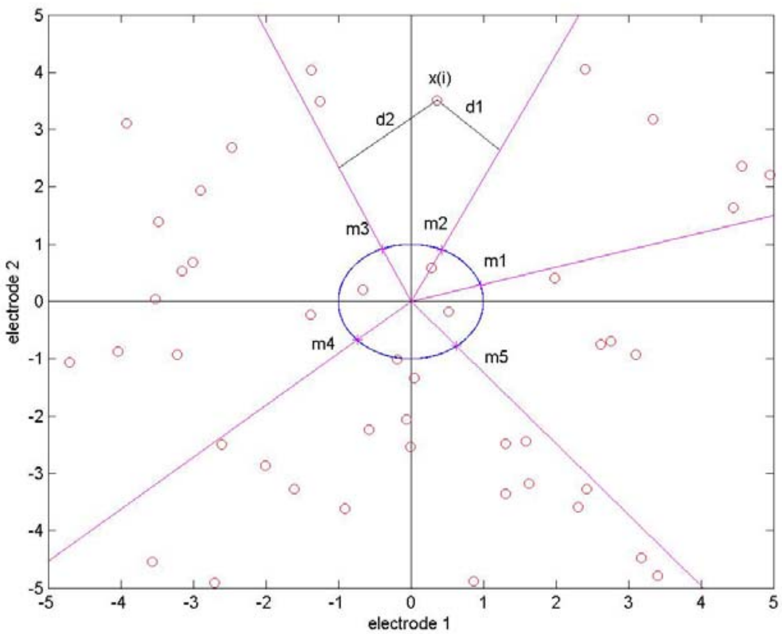


Figure 7: Labeling the data.

Mathematically, the orthogonal squared distance between each measurement and each microstate is computed as

$$d_{kt}^2 = V_t' * V_t - (V_t' * T_k)^2 \tag{4}$$

For more details look at [1] in the appendix.

Labeling the measurements matrix means that a point will be represented by a microstate with certain intensity. Mathematically:

$$L_t = \arg\min_k \{d_{kt}^2\} \tag{5}$$

Where L_t is the vector containing all the labeling for each time t.

2.2.3. Estimation of the parameters T and A:

Approximating our measurements with microstates includes the apparition of an error for each point's estimation. Using recurrence, the algorithm will try to reduce the error converging to a minimum of the function (3). We see that we are in front of a minimization problem (optimization problem). Our goal is to represent a set of n vectors with k microstates, k<n.

This kind of problem has been resolved with k means clustering algorithms used in image compression. The MSS is a derivation of these.

Once we have labeled all the data, we can define an estimator for the non zero a_{Kt} where $K = L_t$:

$$\hat{a}_{Kt} = V_t' * T_K \tag{6}$$

A is the intensity matrix. More details in the appendix [2].

We still have to find the vector T minimizing (3) under constraints that T must be a normalized vector. We are going to use in this objective a correlation matrix (7). This matrix tells us about the correlation between the different axes of the problem. The T microstates are obtained as the eigenvectors corresponding to the largest eigenvalue of the matrix:

$$S_k = \sum_{t \in L_t = k} V_t * V_t' \tag{7}$$

So

$$T_k = \arg\max_X \{X' \cdot S_k \cdot X\} \tag{8}$$

With the constraint $\|X\| = 1$ as we are searching for normalized microstates.

Finding the eigenvector with the highest eigenvalue will always give us the optimal microstate T with norm equal to one maximizing the non linear equation (8). See the proof and more explications in the appendix [3].

Instinctively, we can represent the problem in a simplest way. We have one microstate vector representing, say three points measures (Figure 8). There is an error because it is an approximation.

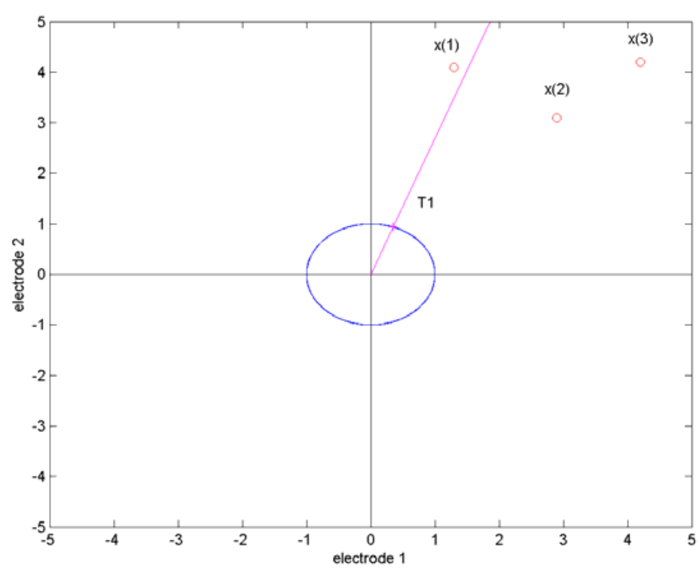


Figure 8: a set of points labeled with T1.

We are searching for a new vector better representing our point's values. We can at first suppose that the next microstate vector should point the center of gravity of the measured points to minimize the error (Figure 9). This solution is good but we are still forgetting an important clue. We are not considering the density repartition of the points. This is why the correlation matrix is used for (Figure 10).

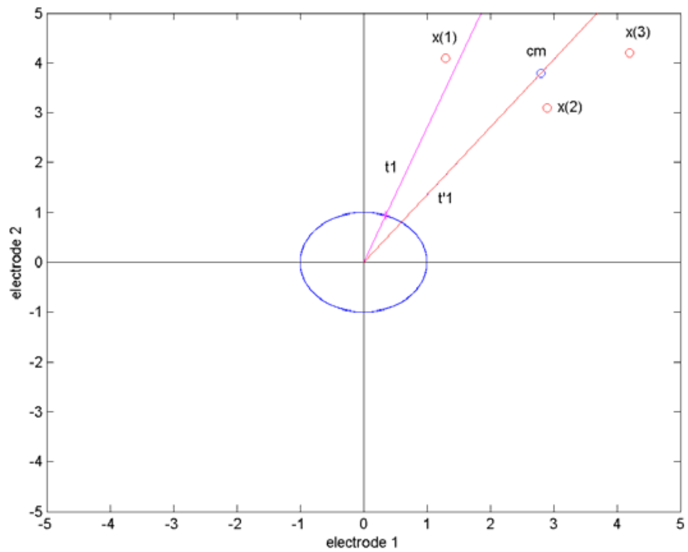


Figure 9: computing the new microstate using the center of gravity.

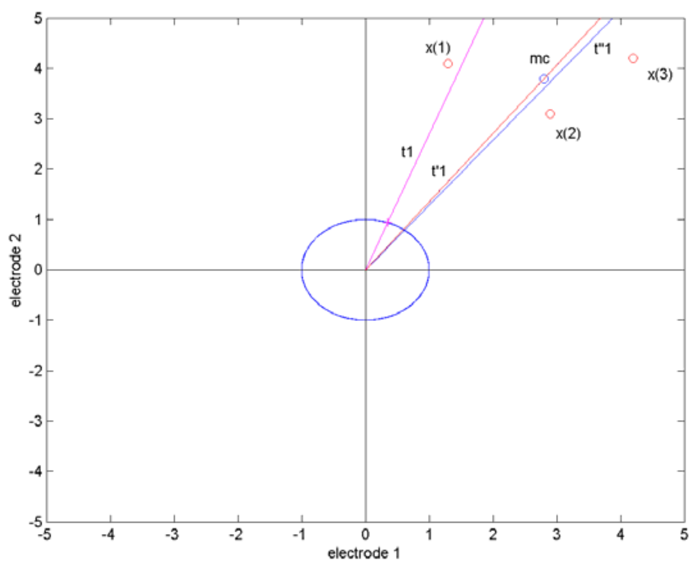


Figure 10: computing the new microstate using the MSS algorithm t_1''.

2.2.4. Convergence criteria:

When can we decide if the algorithm has done enough iteration to converge to a good result? We will look at the error function and decide for a limit to satisfy. This threshold is defined subjectively as $\varepsilon \cdot Err$ with $\varepsilon = 10^{-06}$.

When the error is smaller than this bound we can stop the algorithm and estimate that it has found a good minimum. Otherwise, we should try again iterating.

2.3. Estimating the number of microstates:

In the previous section, we supposed that the number of microstates was known a priori. However, it is never the case and the problem of estimating the number of microstates is important because it has lot of influence on the quality of the results. At first sight, we can immediately see that the best solution is to have as many microstates as points to be represented. In this case the error is null. But practically it won't be feasible and really impossible to implement (there are too many points). We will use a method based on cross validation to determine the optimal number of states N_μ.

Cross validation methods consists of dividing the data into k subsets of (approximately) equal size. We train the net k times on the MSS algorithm, each time leaving out one of the subsets from training (the validation subset) and obtaining the matrix *T* and *A*, but using only the omitted subset to compute the error.

As the microstates vectors are calculate for other subsets, the error will have a particular comportment. For a small number of states, the error will vanish at the same speed than the training set but the number of states increasing, its decreasing speed will tend to be null. At the same time the number of unused microstates by the validation data will augment significantly plotting an increasing curve because of the fact that the matrix T is representing another set of data with more and more precision.

Adding these two results on the same plane we will obtain a convex curve, called the error function. The minimum of this function will give us the ideal number of states to be used. As we are repeating this procedure with different subsets, we will finally obtain a result with error approximation.

So, the error is defined as:

$$Err_{cv} = \sum_{t=1}^{N_t} \left\| V_t - \sum_{k=1}^{N_\mu} a_{kt} \cdot T_k \right\|^2 + E_N \tag{9}$$

Where E_N is the number of microstates unused by the validation set of data. E_N is an increasing function. Indeed, as we are estimating our set of training values with more microstates, these microstates have a bigger probability to be unused by the set of validation data.

The error will be for $t \in [1;128]$ as follow: (Figure 11)

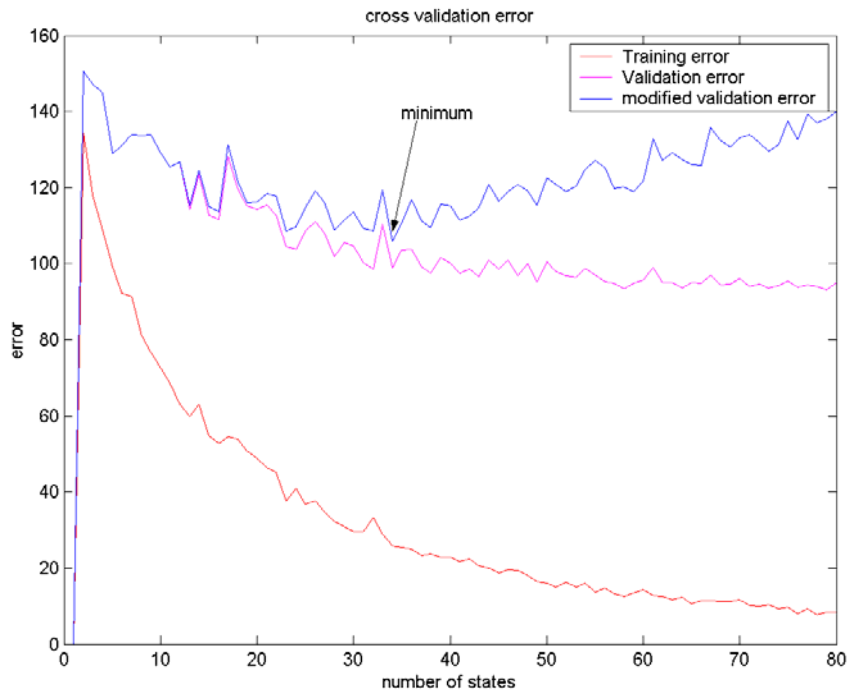


Figure 11: cross validation error function.

It shows us that the best value to take for the number of microstates N_μ is the minimum of the error function. We can ask ourselves why not just looking at the number of unused microstates instead of computing all the errors. It would be an interesting gain of time but unfortunately, the unused curve is not strictly increasing and has not an immediate influence on the error. We are focusing on when it changes the error direction. For a general application, it is useful to observe the schema on the next page presenting the cross validation algorithm (Figure 12). It points out the fact that we are using recurrence on subsets of data and that for the final result, we will have the optimal number of states with a confidence interval. We take the median for deciding which of the different number of microstates found we are going to use.

To have more explanations about the algorithm's implementation, the appendix [4] explains how it has been done in Matlab.

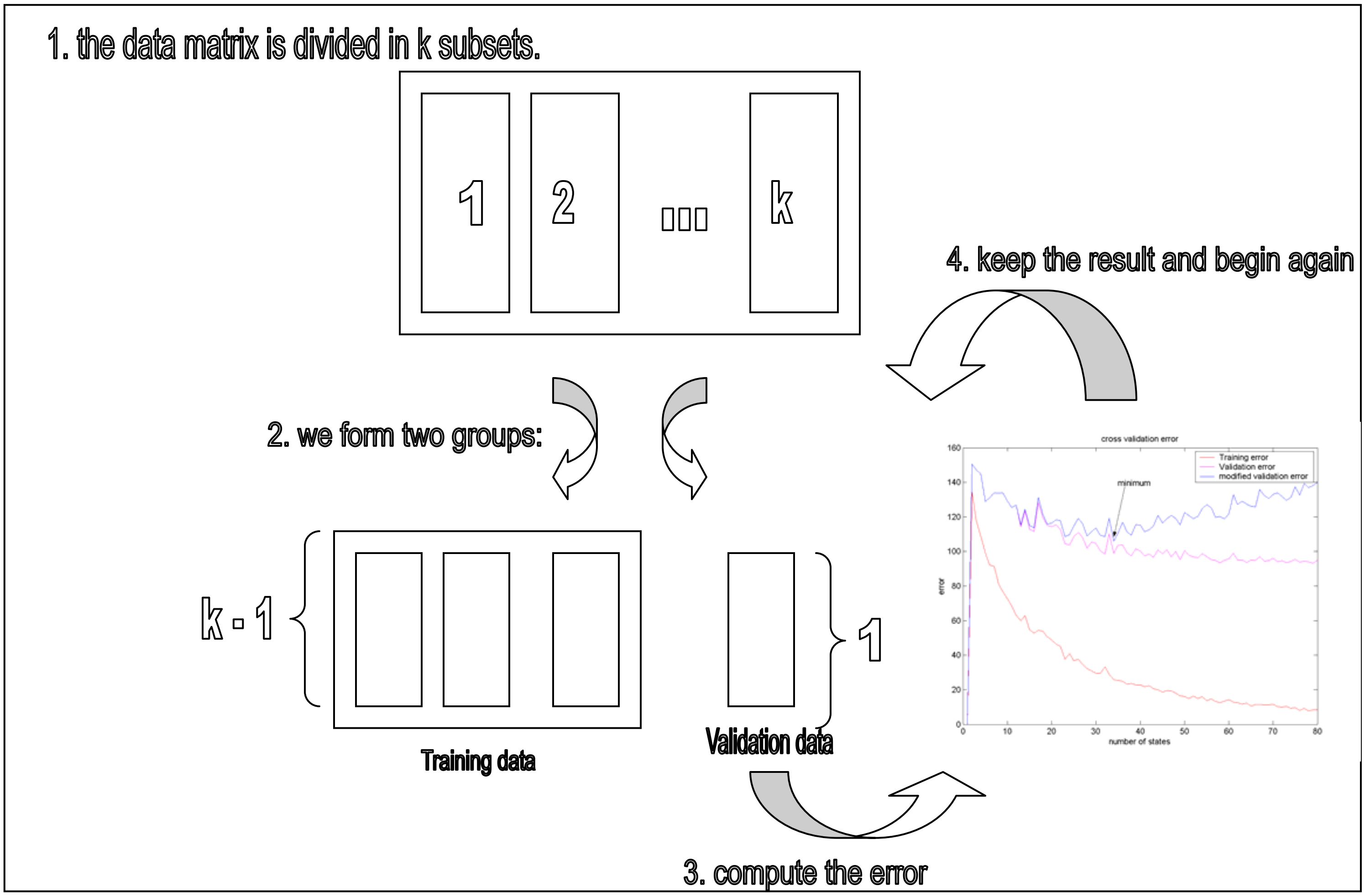


Figure 12: Block diagram of the cross validation method.

2.4. Remarks on the application and difficulties encountered:

2.4.1. Data pre-processing:

Before applying the algorithm, it is important to treat the data for erasing the different noises. When we measure the electrical activity of a patient, high frequency noises enter in consideration [3].

First, the EEG system measures the electrical activity through very sensitive electrodes disposed on the scalp surface. In this way, we measure the electrical variations but also noise due to eye blinks or head movements. For example, when the subject is moving, small vibrations are detected by the electrodes and need to be taken away. This is done by observing the data, and choosing by hand the part we will analyze. We could imagine a system filtering the data using Fourier transforms or any others time-frequency transformation and deleting the unwanted frequencies.

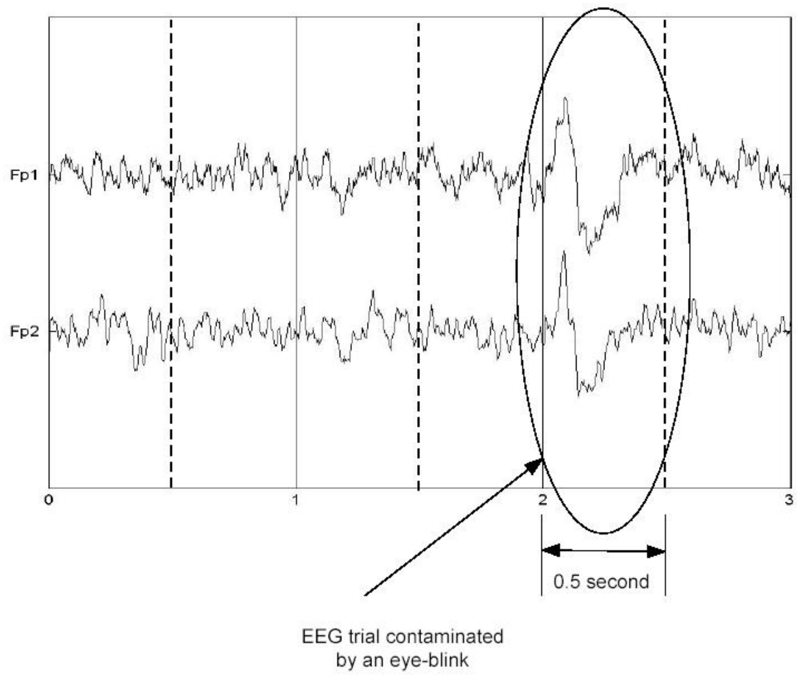


Figure 12: Eye blink detected on the front electrodes and that need to be avoided.

Second, as the perfection is just a theory, when we measure, the electrodes are not only recording potential values but also noise due to perturbations in the room (voices, electromagnetic waves due to metal pieces, high frequency noises...). A good way of resolving this problem is to use the Wavelet denoising. Wavelet transformation will decompose the signal in frequency bands from low frequencies to high frequencies. Deleting the highest frequency decomposition and transforming again in the time domain will give us a denoised signal. A good implementation for our problem is the "minimaxi" [5] that is already implemented in Matlab (*wden function*).

Applying as an example this algorithm on a set of measured data MA1, we can compare the de-noised version with the measured one (Figure 13).

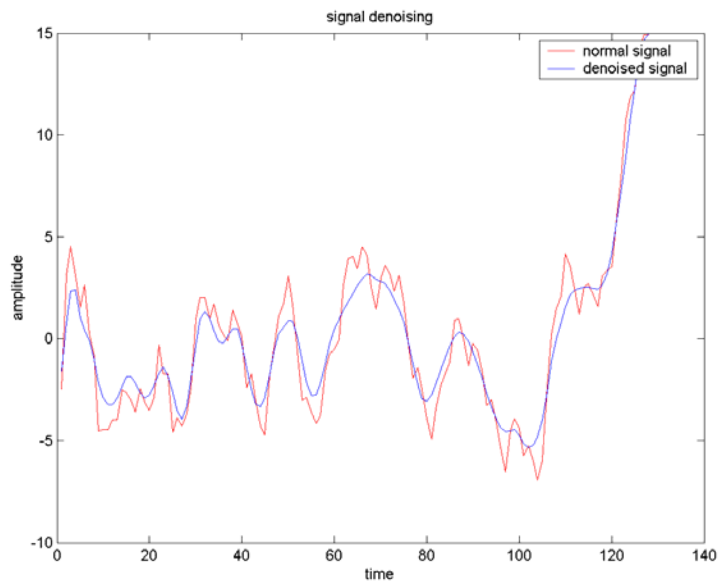


Figure 13: Example of denoising.

Finally, observing a set of EEG data, it should be clear that the time mean will be null. If it wasn't the case, the brain would be an electricity generator! Moreover, in space, the mean should be null as well. It is less evident to observe but adding small electric variations all around the brain, we can conclude that the small variations sum is null all around the scalp (Figure 14). In practice, we obtain a time mean null but not the space mean. It is due to the noise and has to be avoided. Retiring the spatial mean to set of data will give us a better signal to analyze.

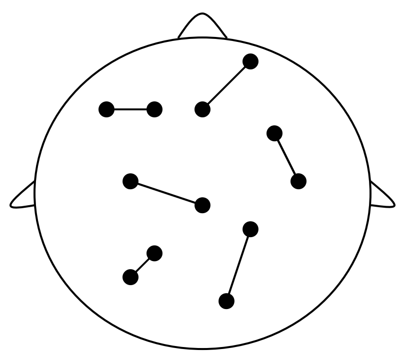


Figure 14: The sum of all electric variations over the head is null.

2.4.2. Initialization problems:

No matter the initialization, the algorithm will converge to a minimum. However, it will not guarantee to find the global minimum but maybe just a local one. Therefore, in practice, it is recommended to restart the algorithm several times with different starting points (the initials T vectors have an important influence on the result!).

The initial microstates are always computed selecting randomly a point in the data matrix and normalizing it. This is the best method since it avoids choosing a microstate that is not used to represent the data and that would be ignored by the algorithm giving unexpected results. However, sometimes it still appears that some microstates are unused. It means that no points in the EEG data are labeled by them.

How could appear this problem if we have taken the precaution to select our initial microstates in the set of data? There are still special cases that cannot be avoided. For example, new microstates can exactly be symmetric to one another and so express a measured point with exactly the same error (d is the same and the intensity values are simply opposite). The algorithm will choose between these two microstates and maybe one of them will never be chosen. In this case, it is a forgotten microstate (Figure 15). It needs to be erased or replaced by other better placed microstates. The problem can also appear in symmetry (Figure 16).

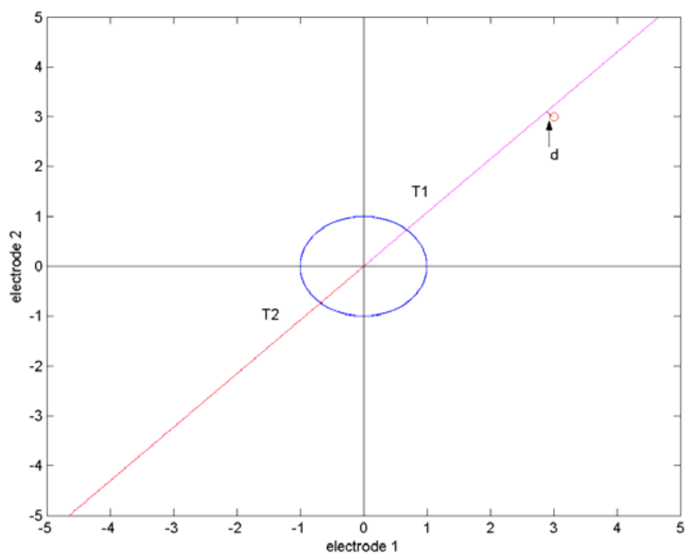


Figure 15: Two microstates equivalent.

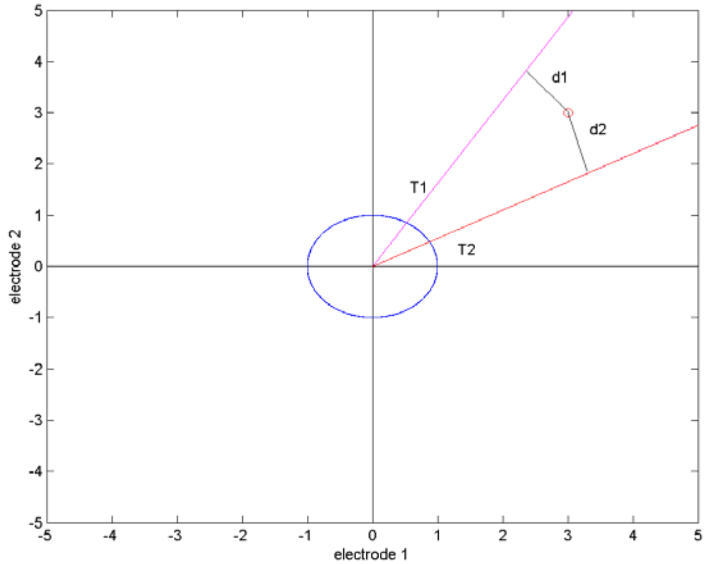


Figure 16: Two microstates with d1 = d2.

To solve the problem, I added a test in the algorithm verifying if there were null microstates. In these particular cases, the best solution was to replace them by microstates pointing out regions with important error. Remember, our goal is to reduce the error between the EEG data and the microstate representation. Following this idea, the best place for a new microstate is where the biggest error is. So we compute the error for each point and select the maximum. The new microstate is then replaced in this direction.

With this method, the algorithm stays convergent and avoids an unexpected response. We can dissert about the fact that if the algorithm find a null microstate, it is maybe because this microstate is not needed and that we should change the number of used microstates. However we have seen that the starting point was really important for the algorithm convergence quality. If a microstate is found null after some iteration, it does not mean that this microstate is useless. It just tells us that its initial position wasn't well defined.

2.4.3. Analyzing the results:

Cross validation error function

Defining the error function for the cross validation was also an interesting challenge. At first sight, the error function should just take in account the difference between the EEG data and its approximation. If we use the same error function than for the MSS algorithm the curve error of the validation set won't be convex but stay constant for important number of states. It is indeed because of the fact that when the number of microstates increases a lot, they won't be used by the validation set of data. These microstates are pointing the training values and are not influencing the error of the validation measurements. As the number of state is increasing, the error for the validation data cannot increase as well but will rather be constant. Increasing the number of states cannot make the error augment.

The solution taken for resolving this problem was that of regarding the number of unused microstates that had to increase respectively with the number of microstates. This was a good solution permitting us to obtain a convex curve for the error.

Code optimization

The algorithm is quite slow when it has to deal with important samplings. It needed a lot of time to converge to the solution because of its important number of sums and recursions. There aren't many solutions to this kind of problems. Either we change our method of application (by compiling the algorithm and using just its machine code) or by having a look to its slow instructions. There was an important work of code optimization in order to improve significantly the quickness of the convergence. This point is very important as the algorithm will be used in real time applications! Before guessing the person thoughts, we need to compute the model and this should not be too long not to exasperate the user!

Another way of optimization is to try to avoid some calculus. Sometimes, it is possible to find a method giving satisfying results in less time. A parallel method was found for the algorithm computing the best number of microstates (cross validation).

In this algorithm, we compute the cross validation on a set of data, changing the number of states. Each time it is needed to compute the initials microstates to run the algorithm. An interesting option is to keep the microstates resulting from the previous iteration and to add just one new microstate. With this method, we avoid having to compute again the initialization and this way we obtain a faster convergence.

We observe that with this method, the results are found two times faster than with the usual one. However, this gain in time is followed by a more important error on the determination of the optimal N_μ. To determine if this particular method is effective, it is important to prove that it gives us results as good than those found with the normal method. With the following results, we can conclude that this method is efficient enough to be used when dealing with important sets of data. This is an interesting way to take in account to gain time in the execution of the algorithm. For a signal length of 300, we obtain:

	Method 1 (slow)	Method 2 (quick)
Time [sec]	71.8110	42.5150
Number of states	31	26
Confidence interval	3	7

3. Results:

An important test for the algorithm efficiency is to compare its results with another one. We are going to use Kernel Based's algorithm (KB) well introduced in reference [6]. Both algorithms are implemented in Matlab.

The MSS is currently used in the medical world and has given good results until now. KB's algorithm has proved its capabilities on different projects developed by the Swiss Federal Institute of Technology. It offers at present the possibility to recognize four different mental tasks and to give a feed back to the user in a relative small among of time.

3.1. Recognizing the data:

Our first test will consist of determining either an unknown set of recorded data owns a known mental task. Each algorithm will compute its solution. In the first result (Figure 17), we tested the algorithm the following way: We computed for each model MA1, MA2 and MA3 their microstates respectively T1, T2 and T3. Say we take the model one in account; the error should be minimum with the EEG signals of the model one when we apply T1 on MA1, MA2 and MA3. Representing the results in a graph, we will found out that the minimums are on the diagonal since for each model the smallest error will be computed for its corresponding EEG signals.

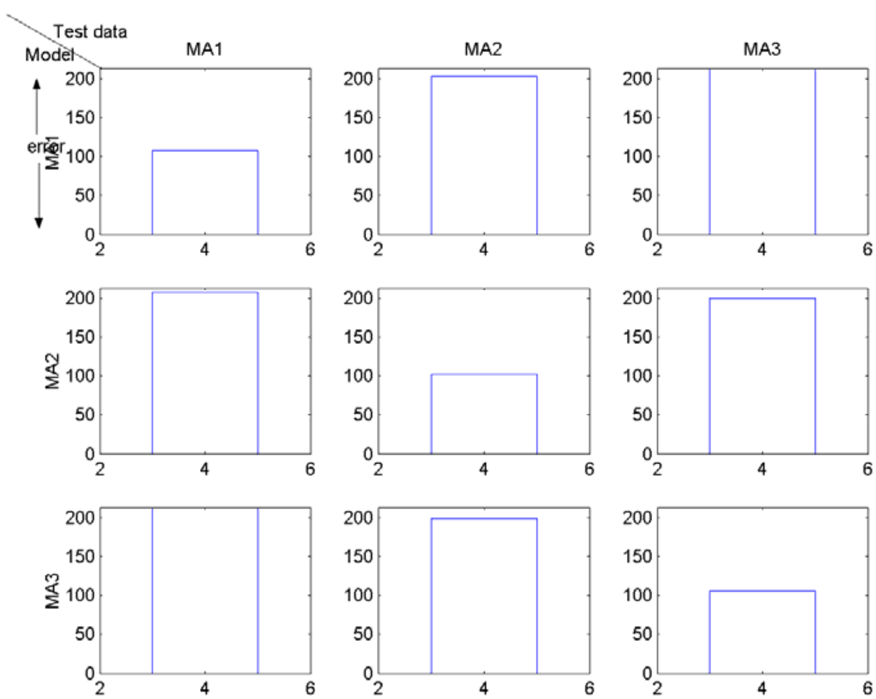


Figure 17: Testing the algorithm.

The next objective is to apply the algorithm to a pattern recognition problem. We are going to define a procedure for being systematic and be able to analyze well the results.

Procedure:

We suppose we have three set of data MA1, MA2 and MA3 representing three different tasks. In the first step, we will use MA1 as the model and compute the errors for each set of data. Plotting the error in a histogram will exhibit whether or not the algorithm manages to differentiate the different tasks. Ideally, the results will be of the form of the figure 18. In practice, we will rather have something like figure 19.

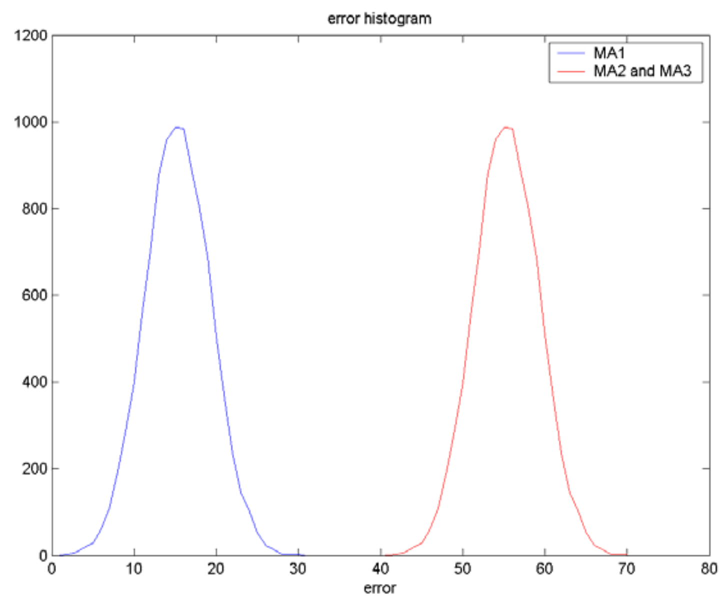


Figure 18: theoretically.

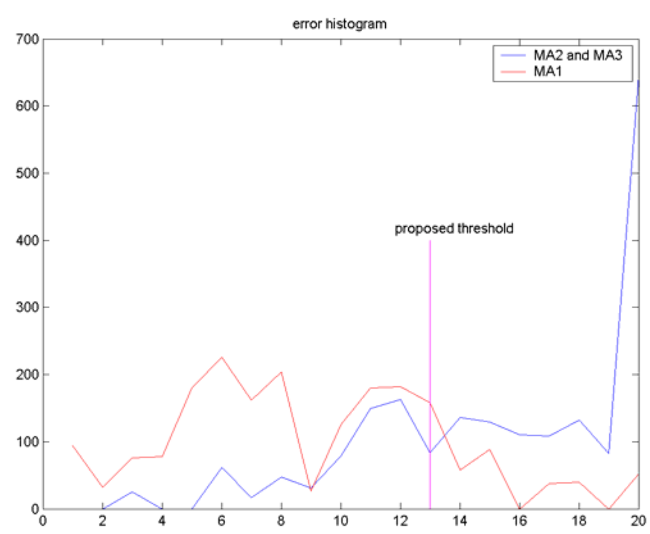


Figure 19: in practice.

This procedure is restarted several times using then MA2 and MA3 as model.

We still need to compute a threshold that will help for doing the right choice. We can see on the figures above that the threshold should be in the middle of the two maximums. Moreover, there exist different policies of error acceptation. If we move the threshold on the right, we are going to augment the error (doing a false choice) but augment the feed back for the user. Moving the limit to the left, will contribute in having more exact answers but with a feed back less efficient… (Figure 20)

The threshold is defined as:

$$K = \left| \frac{a+b)}{2} \right|$$

where

$$a = \textit{Position of the maximum of the red curve.}$$
$$b = \textit{ Position of the maximum of the blue curve.}$$

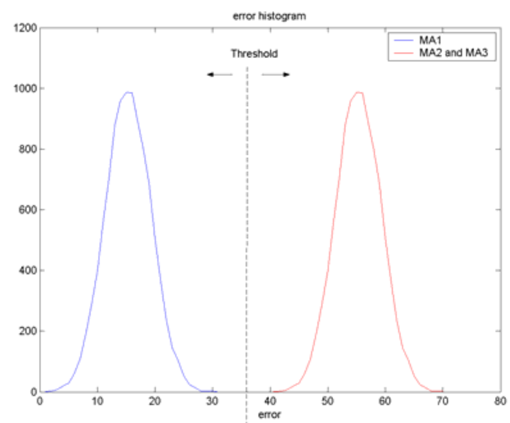


Figure 20: threshold.

3.2. Comparing the results:

To establish the comparison, the False-Positive and False-negative language will be employed. Applying the procedure defined above, we obtained the following results:

False positive False negative	MSS	Kernel based
MA1	21% 16%	12.3% 5.4%
MA2	36% 17%	26.2% 7.8%
MA3	35% 6%	13.8% 3.7%

Before analyzing, we have to precise that the Kernel Based (KB) algorithm results were found using a relaxed threshold; in other words, KB admitted easily errors and remembering the figure 21, the threshold had been moved on the right. This is why the False-negative are so close to zero.

To obtain the previous results, we have to be aware that it needed an important amount of time. The longest part consists of building the model. This is done depending on the size of the data, in approximately one minute to half an hour. The results are positive since they prove that the algorithm efficiency is good enough for determining with an acceptable probability the thought of the user. We can note that the obtained results are not as good as with the KB algorithm.

Analyzing the two ways of resolving the problem, we denote important differences. The MSS algorithm does not use the probabilistic side of the problem in account and misses lots of information. The user is not able to keep a mental task stable for a long time; its concentration failure will generate unwanted states that have to be ignored! Moreover, the brain electrical activity depends on the user physical state at the moment and could influence in a negative way on the results.

3.3. Algorithm complexity:

To measure the efficiency of the algorithm, the big O analysis is helpful to compute its time and space complexity and to try to improve the algorithm possibilities. The big O analysis determines the algorithm convergence and exhibits its ability to treat important sets of data.

For n being the EEG data size (time dimension), the complexity is found to be linear, that is $O(n)$. It is just an observation of the time needed to have the result changing the size of the entry data. It is a very interesting result about the algorithm convergence.

EEG signals length	Execution time [sec]
128	21
2*128	38
3*128	59
4*128	78
…	…

The complexity is interesting information for determining the algorithm capacities; it helps for determining the time to solve a given problem.

4. Conclusion:

Is the algorithm efficient enough to consider using its results as a basis for next research? In the actual states of knowledge of the brain functionalities, it is not yet possible to totally refute or integrate an analysis method. Today this method can be good enough for our needs but maybe it will be replaced by more exact methods in the next years. However, it is undeniable that the MSS algorithm is not taking in account some very important parameters of the problem. The brain is never working with the same intensity; it really depends on the subject physical state at the moment. Moreover, when we are sampling the data, the subject is not always disposed to concentrate himself on a particular mental task, and is not able to keep in mind the same idea constantly. These limits exhibit the algorithm weaknesses, and show us new ideas to explore.

Taking in account the probabilistic side of the problem would be an important evolution. In the MSS algorithm, it is certainly the biggest weakness. Trying to elaborate a more sophisticated way to label the data rather than just computing the Euclidian distance or develop another method of data recording would also be a consequent amelioration.

However, we still can ask ourselves about the limits we should not cross. Are we going to transform the human in a machine, exploring one of its last mysteries? Or at the contrary, are we going to develop our knowledge of our brain for the health of the humanity?

Future perspectives?

Let’s remember that there are many future applications from the medical world to the educational world. As for the example given in the introduction, presenting to us a way of replacing a damaged nerve intercepting the brain activities (Figure 21), we can image lots of different applications on the human body. These discoveries are, surely, temporary since we can expect the biologist to develop better way to redo our own bodies, particularly by regenerating our own cells. Another medical application would consist in using this new technology to help people suffering from epilepsy to control their own brain as a muscle. This idea reveals an important horizon offered to us by this technology.

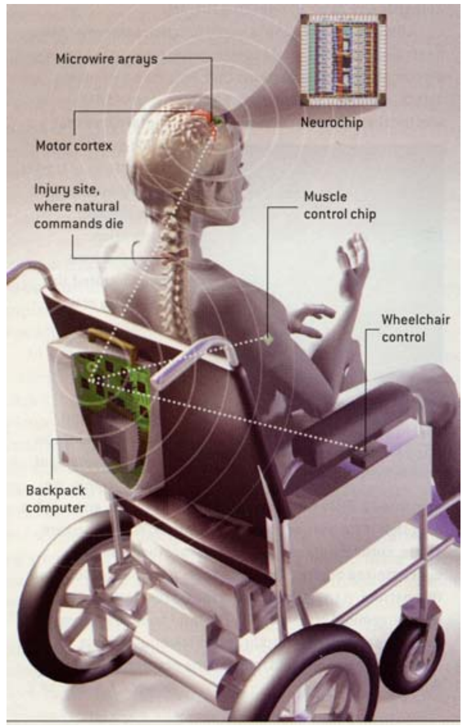


Figure 21: Example of future application [9].

PhD. B. Pert in her thesis about the embodiment of the mind [7] explained that: ”The body reflects the unconscious of the brain”. What she meant by this sentence is just a supposition but offering immense perspective. We often forget to listen to our bodies and our feelings, we contain our brain natural expressions and the only way he managed to express himself is through the body. Our unconscious is going to liberate these ignored feelings damaging its own cells, or developing body’s illnesses.

Developing a brain-computer interface, we are trying to listen to the brain, the origin of all our feelings [2]. But we could imagine continuing to improve our communication with computers, developing interfaces based on the sight, the touch and so on. The computer will in this way transform itself in a multi-sensory computer (MSC). Perceiving all our emotions and feelings, a computer becomes human and would be able to tell us about our feelings, our mental states, our unconscious, and help us to understand ourselves better. The MSC would tell the human how he fells.

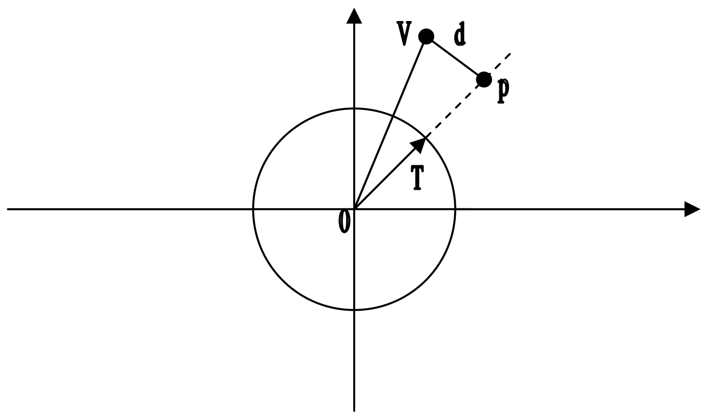
We should ask ourselves why creating a human machine when there are so many human beings in the planet. Maybe should we consider building a human, to begin understanding ourselves, and finally begin to communicate each others?

In studying our brain, it is a new way of reunification of the human corps with its brain. Accepting our feelings, listening to them is a therapy to discover ourselves and to liberate the human from his fears. Studying the brain is the only way to discover one of the mysteries of the human being, trying to determine ourselves, our feelings to improve our conditions of life. A recent study on population, exhibit that there were more and more people using medicine to relax themselves and permit them to continue living their life normally [8]. These medicines work on the brain connection making them weaker to diminish their influence on the body. Instead of hiding our own brains we should maybe enter in a new way of thinking in trying to establish a better understanding of ourselves.

Appendix:

[1] ***Proof:***

We are going to show how is computed the distance d between the measures and the microstates.



If we have a look on the figure above, we can immediately notice that it d can be expressed using the Pythagoras's theorem as:

$$\vec{OP}^2 + d^2 = \vec{OV}^2$$

This can be rewritten as: $d^2 = \|V\|^2 - \langle V, T \rangle^2$

Where $\|V\|^2$ *is the square norm of V and* $\langle V, T \rangle$ *is the projection of V onto T using the scalar product notation.*

Remembering that: $\|V\|^2 = \langle V, V \rangle$

And

$$\langle a, b \rangle = \langle b, a \rangle = b^T \cdot a$$

We find:

$$d^2 = V_t' \cdot V_t - (V_t' \cdot T_k)$$

□

[2] ***Proof:***

Let's prove of to find the matrix A of microstates intensity. Using the same drawing than for the proof [1], we see immediately that the intensity of the microstate representing V must be equal to the projection of V onto T. Mathematically, it is done by the scalar product; i.e. :

$$\widehat{a}_{kt} = \langle V_t, T_k \rangle = V_t' \cdot T_k$$

□

[3] ***Explications about the mean correlation matrix:***

The correlation matrix expresses the relationship between the axes of the dimension in which we are working. Let's take a simple example to introduce the idea.

Remember the case we took to introduce the algorithm, considering just two electrodes and so working in a 2-D plane. If we have three points represented by microstates, we would like to determine the next microstate best representing the points; when the global error is minimum.

We have:

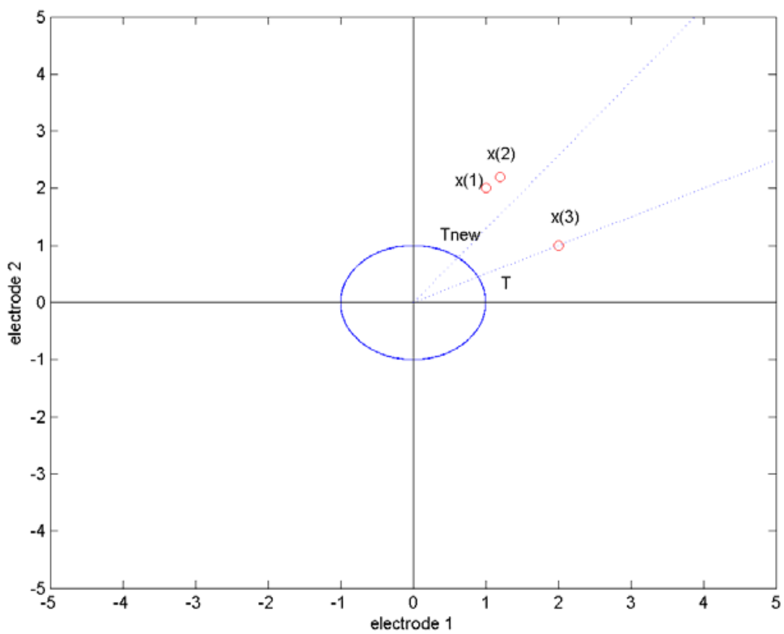
$$x(1) = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$x(2) = \begin{bmatrix} 1.2 \\ 2.2 \end{bmatrix}$$

$$x(3) = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

and the microstate:

$$T = \begin{bmatrix} 0.6 \\ 0.3 \end{bmatrix}$$



Just by having a look on the figure, we can see that the best place for the new microstate would be near the two points that are close together. There the error would be small. But how to express this natural fact mathematically?

With these values the correlation matrix is:

$$S = \begin{bmatrix} 6.44 & 6.64 \\ 6.64 & 9.84 \end{bmatrix}$$

It means that on the x axis and on the red and blue axis on the figure, the correlation is quite identical. However, the value for the y axis tells us that the next microstate should have a y coordinate a little bit bigger. The method used here writes this observation as:

$$\begin{bmatrix} T_1 & T_2 \end{bmatrix} \cdot \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \cdot \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} = S_{11} \cdot x_1^2 + S_{21} \cdot x_1 \cdot x_2 + S_{12} \cdot x_1 \cdot x_2 + S_{22} \cdot x_2^2 = \max$$

This way, the vector T obtained will reflect as well as possible the correlation of the points and diminish the error better than just pointing the mass center!

We still need to prove that the best solution for this kind of equations is always given by the eigenvectors.

Proof:

Let's take a vector y with norm equal to 1. We are going to prove that the best solution we can find will always be smaller or equal to the eigenvector x with the biggest eigenvalues. We will do the demonstration with a two dimension example, but the generalization will be straight forward.

$$\begin{aligned} & y' \cdot A \cdot y \le x' \cdot A \cdot x \\ & y' \cdot A \cdot y = y' \cdot (P' \cdot D \cdot P) \cdot y \\ & so: \\ & (P \cdot y)' \cdot D \cdot (P \cdot y) \le (P \cdot x)' \cdot D \cdot (P \cdot x) \end{aligned}$$

We know that:

$$\begin{aligned} A \cdot x &= \lambda \cdot x \\ P' \cdot D \cdot P \cdot x &= \lambda \cdot x \\ P \cdot x &= D^{-1} \cdot P \cdot \lambda \cdot x \end{aligned}$$

Using that P is orthogonal: $P' = P^{-1}$

We find:

$$(P \cdot x)' \cdot D \cdot (P \cdot x) = \lambda \cdot x' \cdot P' \cdot D^{-1} \cdot D \cdot P \cdot x = \lambda$$

The problem becomes to demonstrate that:

$$(P \cdot y)' \cdot D \cdot (P \cdot y) \leq \lambda$$

D is diagonal:

$$D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

Let's say:

$$P \cdot y = z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

We can write:

$$z' \cdot D \cdot z = z_1^2 \cdot \lambda_1 + z_2^2 \cdot \lambda_2$$

With constraint:

$$z_1^2 + z_2^2 = 1$$

As:

$$\lambda_1 < \lambda_2$$

We pose:

$$r \cdot \lambda_1 = \lambda_2 \ r \in [0,1]$$

So:

$$\lambda_2 \cdot (r \cdot z_1^2 + z_2^2) < 1$$

This expression must be smaller than one. It is always the case remembering the constraint on r *and* z_1, z_2.

[4] ***Implementation:***

All the manipulations have been executed on Matlab from the Mathworks Company. Matlab is a powerful software when resolving mathematical problems. Three functions have been implemented.

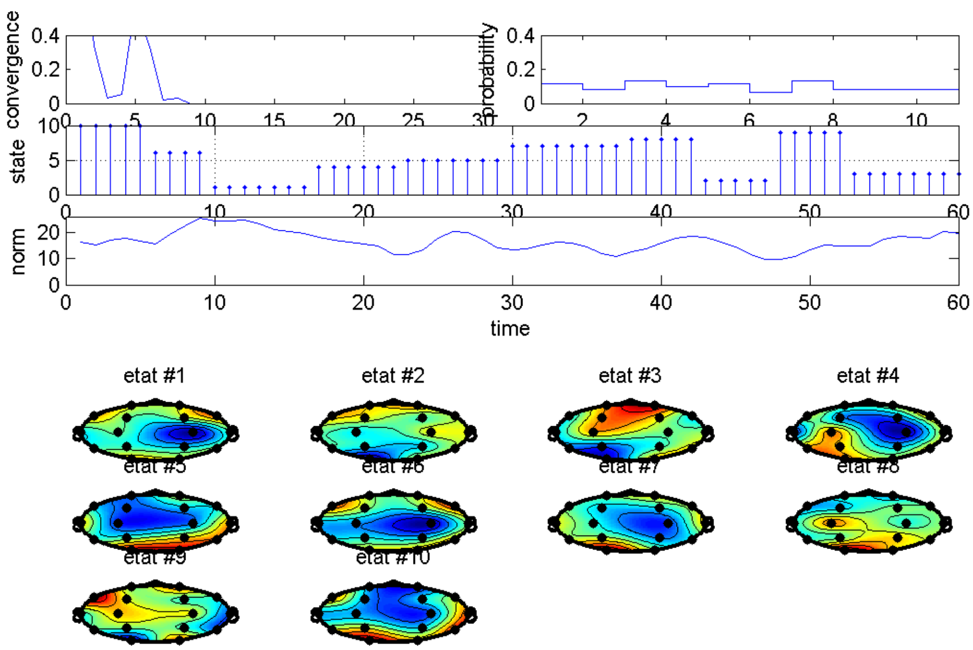
mss_f_state function:

This function computes the MSS algorithm on a set of given EEG data. The results are plotted in a figure as follow:

At the top left, we can see the algorithm error tending to zero. It gives us the information about the number of iterations that had to be done before having a satisfying result. At the top right, the graph tells us the probability of being in one particular microstate. If there are ten microstates for example, it will tell us that we have 1/8 probability to be in the third microstate in this set of data.

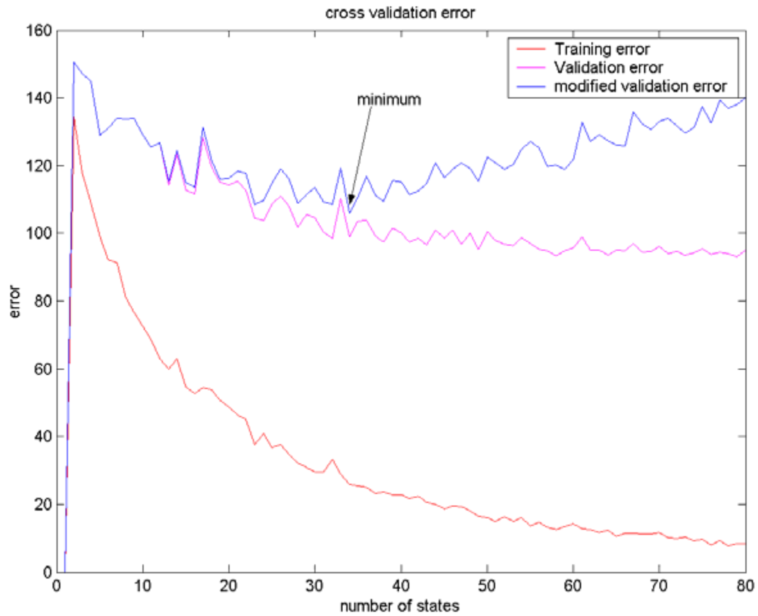
The second part of the figure shows the evolution of the labeling during the time. We can observe that sometimes, the microstate is stable for a while. The third part tells us about the amplitude of the norm of the set of data at each time t. It is an interesting information because we can observe that when we are in a minimum, the probability of a change in the microstates labeling is higher than for others time instants. It can be very useful in case of very important sets of data. Instead of studying all the measurements, it tells us about which are the interesting parts we should consider with more attention.

The fourth part exhibit the scalp map configuration of each computed microstate. It is just visual information to observe the evolution of the more active part of the brain.



f_mss_nb_state function:

When we want to know for EEG measures which is the optimal number of states for representing the set of data, this function will do that for us. In fact, Calculate computes one step of the correlation method. If we would like to use the cross validation method we have to use recurrence and call several time the Calculate function. The result is the optimal number of microstates and a plotting of the error graph. This graph exhibits a convex curve where the minimum represents the optimal N_{μ}.



f_mss_threshold function:

This function computes the threshold given a model, its corresponding set of data and two others EEG signals matrix. It is based on the previous two functions. You have to take care on which data you will compute the threshold. If you obtained the microstates for, say MA1_d(:,:,1:10), you should apply the f_mss_threshold function on the same set of data! Otherwise the results won't be analyzable.

f_mss_class function:

The goal of the algorithm we have been talking about is to determine the thoughts of a subject using a computer environment. This is the function to be used in that purpose. For, say, three set of EEG recordings representing respectively, the left hand movement, the right hand movement and the mental activity consisting of counting Compare will compute the set of microstates for each recordings and use them to identify EEG measurements as belonging or not to a particular mental activity.

References:

[1] G. Garcia, T.Ebrahimi and J.-M. Vesin, “Joint Time-Frequency-Space classification of EEG in a brain-computer interface application”, *to appear in the Journal of Applied Signal Processing (special issue on neuromorphical signal processing)*.

[2] T.Ebrahimi, J.-M. Vesin, “Brain Computer Interfaces for multimedia communication*”, IEEE Signal Processing Magazine*, January 2003.

[3] Gary N. Garcia, T. Ebrahimi and J.-M. Vesin, “A direct brain-computer communication device with user rewarding mechanism”.

[4] R.D. Pascual-Marqui, et al. “Segmentation of Brain Electrical Activity into Microstates: Model Estimation and Validation”, *IEEE Transactions on Biomed. Engineering*, vol. 7, no. 7, 1995.

[5] S. Mallat, “Wavelet tour of signal processing”, Second edition, Academic Press, 1999.

[6] Gary N. Garcia, T. Ebrahimi and J.-M. Vesin, "Human-Computer adaptation for EEG based communication", Submitted to the IEEE EMBS conference 2003.

[7] PhD B. Pert, “The matter of mind: emotions as the mind-body connection“, presented at the school of medicine of Washington DC, USA, 2002.

[8] Jean-Pierre Changeux, “L’homme neuronal”, Hachette Littérature, Pluriel sciences, 2002, pp. 340-344.

[9] Miguel A. L. Nicolelis and John K. Chapin, “Controlling robots with the mind”, Scientific American, October 2002.

The 27th June 2003 Julien Freudiger