Journal of Engineering Sciences, Assiut University, Vol. 40, No 2, pp.557-566, March 2012

MARKOV-BASED BRAIN COMPUTER INTERFACE

A. F. Elqady*, A. A. A. Nasser**, and M. A. Sharkas**

** Faculty of Engineering, Arab Academy for Science and Technology, Alexandria, Egypt*

*** Arab Academy for Science, Technology and Maritime Transports, Alexandria, Egypt*

(Received January 11, 2012 Accepted January 22, 2012)

Although the brain recognizable wave patterns are of limited vocabulary representing finite number of states, the dynamics and moral conditions made it unlimited and subject to severe variations. To enhance the response to certain order, these variations should be augmented in such a way that the brain responds to the nearest meaningful order. In this paper the retention of mental state to the nearest meaningful order is simulated by the retention of nearby periodic orbits out of a chaotic flow. A distributed chaotic generator is used to imitate an intermittent behavior with the laminar phase representing the definite mental tasks and the bursts representing the noisy or undefined tasks. In the intermittent section of the response the nearest periodic response is period three. An algorithm based upon the hidden Markov chain has been developed to retain periodic responses out of the chaotic flow.

1. INTRODUCTION

Brain-Computer Interfaces (BCI) are communication systems which enable users to send commands to computers by using brain activity only, this activity being generally measured by ElectroEncephaloGraphy (EEG) [1]. Most EEG-based BCI are designed around a pattern recognition approach: in a first step features describing the relevant information embedded in the EEG signals are extracted. They are then fed into a classifier which identifies the class of the mental state from these features. Therefore, the efficiency of a BCI, in terms of recognition rate, depends mostly on the choice of appropriate features and classifiers[2]. A good feature extraction algorithm should capture the relevant information related to each targeted brain activity pattern (or mental state) while filtering away noise or any unrelated information[3]. The ability of computers to enhance and augment mental and physical abilities and potential is of great interest. It is becoming a reality after being a dream of science and fiction [4]. One of the important military applications of BCI (Brain Computer Interface) is the instant broadcasting of a command from the commander's brain to the troops under his command. A commander needs only to think of a command to instantly broadcast it to other troops. Fighters also need fast reaction during air raids. This can be achieved using a BCI. The BCI also holds the promise of bringing sight to the blind, hearing to the deaf, and the return of normal functionality to the physically impaired. The brain signal undergoes severe dynamics as it represents unlimited vocabulary. In this respect it resembles a chaotic flow in the vicinity of periodic orbits. In chaotic conditions, the system loses memory to itself (undefined mental task) in the periodic state the system retains memory (a defined mental task). So, if a Markov algorithm can retain periodic flows out of chaotic ones, it can be used to define mental tasks out of unlimited vocabulary.

Figure 1. Distributed Chaotic Generator

The circuit shown in Fig.1 is the distributed chaotic generator. The state equations of which is given by:

$$\boldsymbol{v}^-_1\ (t+T)= v^-_2\ (t) - \boldsymbol{Z}_o\ f(\boldsymbol{v}_1)$$

$$\boldsymbol{v}^-_2\ (t+T)= \rho v^-_1(t)\ , \qquad (1)$$

$$f(\boldsymbol{v}_1)= -\boldsymbol{v}_1 + \boldsymbol{v}^2_1$$

where, v^-_1 and $\boldsymbol{v}^-_2$ are the reflected voltages at the transmission line ends, ρ is the reflection coefficient at the load side, $Z_{0,}$ and T are the characteristic impedance and the delay time of the transmission line respectively, and R_l is the load resistance, Assuming the load is matched and the transmission line is lossless, advancing the state vector one cycle the system in (1) reduces to:

$$\boldsymbol{v}^-_1\ (t+2T)=\boldsymbol{R}_l\ v^-_1\ (t+T)\ (1- \boldsymbol{v}^-_1\ (t+T)) \qquad (2)$$

The above equation is similar to the logistic map [5].

2. CHAOTIC TIME SERIES GENERATION USING THE DISTRIBUTED OSCILLATOR

Using equation (2) with the bifurcation parameter $\boldsymbol{R}_l$ varying from 2.5 up to 4 the dynamics of the system are shown in the bifurcation diagram of Fig.2. The period 3 window appears at the value of 3.827940 for the bifurcation parameter $\boldsymbol{R}_l$

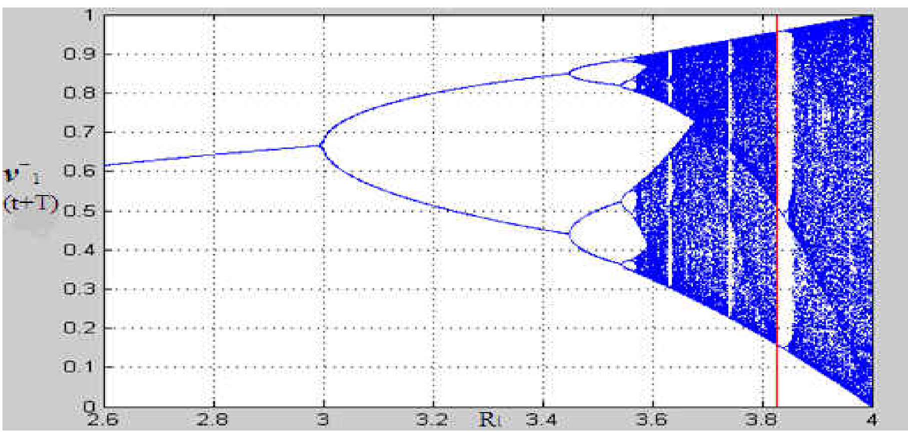

Figure 2. Period (3) at (R_l = 3.827940)

In time domain, with R_l = 3.827940, 100,000 iterations produce the time series shown in Fig.3 where laminar phases interrupted by bursts representing defined and undefined mental tasks.

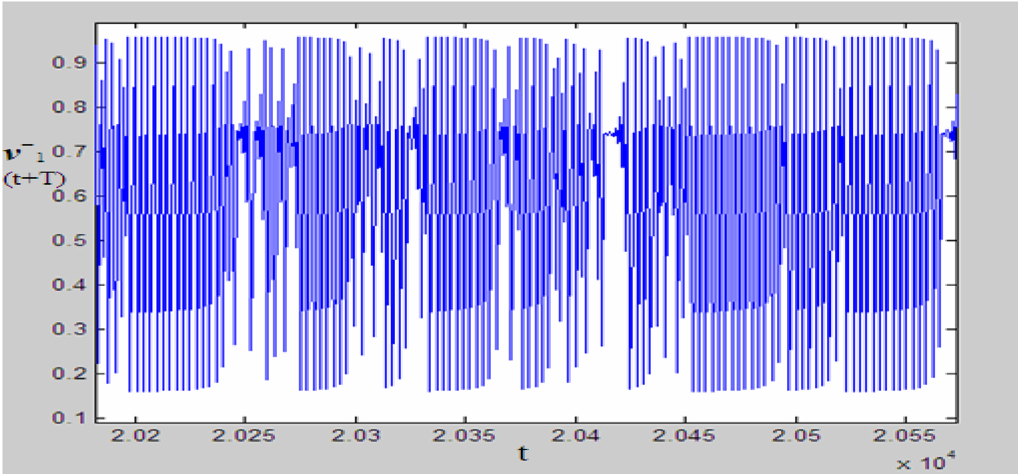

Figure 3. The reflected voltage at the load side with R_l=3.827940

We can distinguish the periodic part from the chaotic part by observing period (3) which repeats the values {0.956 0.160 0.514}. These three successive values may occur one time, in this case it is called period (1) and may be repeated {2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10 , 11 , 12 , 13 , 14 , 15} times , each repeat will be named according to the period[5]. Table 1 shows the period count. The distribution curve showing the probability density of each period relative to the total number of iterates is shown in Fig.4.

Table.1 The count of periods

Period	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Count	1119	526	276	174	124	78	72	76	50	70	50	75	704	126	445

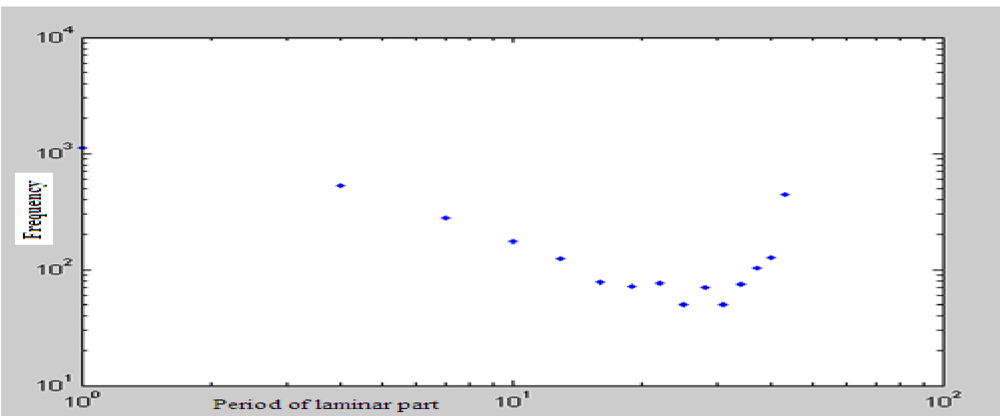

Figure 4 Distribution of periods in the periodic part of the time series.

After counting the probability of occurrence of the periodic iterates (laminar phases), we can compute the probability of occurrence of the chaotic parts (bursts) by subtracting the total count of periodic parts from the total number of iterates.

The count of the periodic parts = 54858 iterations. The count of the chaotic parts = 45142 iterations.

3 . MARCOV CHAIN MODELING

In this section, we model the intermittency chaos by using Marcov Chain. The probability of existence of each period related to total iterations is shown in Table 2.

Table 2 The probability of existence of each period related to total iterations .

Period	1	2	3									13	14	15
Ratio	0.01119	0.00526	0.00276									0.00104	0.00126	0.00445

To calculate the transition probability between the states, we should count the existence of each period related to the other periods is shown in Table 3.

Table 3 The existence of each period related to the other periods .

Period	1	2	3											14	15
Count	0.10095	0.06738	0.05160											0.01713	0.01335

In order to design the Markov model we consider that every period in the periodic part presents one of the states {S1 , S2 , ……. , S15} and all the chaotic parts present one state (S0). Each period has only two probability transitions, one to the next state and the other to the state (S0) except (S15) which has only one transition to (S0) such that:

$$\sum^{N} Q(Sn) = 1 \qquad (3)$$

Where the summation is carried out over the whole length of the sequence. To determine the ratio of the chaotic period we should count the iterations in each periodic period as shown in Table 4.

Table4 Probabilities of periodic and chaotic parts .

Period	1	2	3											14	15	All periodic	All chaotic
Ratio	0.03357	0.03156	0.02484											0.05292	0.20025	0.54858	0.45142

$$Q(Sn) = P(Sn / Sn\text{-}l)\ Q(Sn\text{-}l) \qquad (0 < n \leq L) \qquad (4)$$

From equation (3) we can compute the probability of transitions between states, where (n) is number of periodic states, N is number of total states, Q is the existence of each period related to the other periods and P is the probability of transitions between states.

$$P (Sn+1 / Sn) + P (S0 / Sn) = 1 \qquad (0 \leq n < N). \qquad (5)$$

Equation (5) shows that the probability of transition from the state (n) to the next state (n+1) plus the probability of transition from state (n) when it turns back to chaotic part which presented here in state (S0) equal one. It means that all states have only two transitions, one from itself to the next and the other one when it turns back to state(S0) except (S15)has only one probability that it turns back to the state(S0), All the transition probabilities was calculated from equations (4 and 5) and results shown in Table 5.

Table5 Transition probabilities between different states.

Probability of transition from state (n) to (n+1)	Probability of transition from state(n) to (0)
P0,1 = 0.22363	P0,0 = 0.77637
P1,2 = 0.66746	P1,0 = 0.33254
P2,3 = 0.76581	P2,0 = 0.23419
P3,4 = 0.83953	P3,0 = 0.16047
P4,5 = 0.87950	P4,0 = 0.12050
P5,6 = 0.90236	P5,0 = 0.09764
P6,7 = 0.93194	P6,0 = 0.06806
P7,8 = 0.93258	P7,0 = 0.06742
P8,9 = 0.92369	P8,0 = 0.07631
P9,10 = 0.94565	P9,0 = 0.05438
P10,11 = 0.91954	P10,0 = 0.08046
P11,12 = 0.9375	P11,0 = 0.0625
P12,13 = 0.9	P12,0 = 0.1
P13,14 = 0.84593	P13,0 = 0.15407
P14,15 = 0.77933	P14,0 = 0.22067
-----------	P15,0 = 1

$$Q (S0) = \sum_{n=0}^{N-1} P (S0 / Sn) \; Q (Sn) + Q (SN) \qquad (6)$$

Getting all of the state transition probabilities is shown in Table 6, we can now design the model that describes the time series generated by the distributed chaotic generator.

Table 6 : State transition matrix of state probabilities

	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
S0	0.77637	0.22363	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S1	0.33254	0	0.66746	0	0	0	0	0	0	0	0	0	0	0	0	0
S2	0.23419	0	0	0.76581	0	0	0	0	0	0	0	0	0	0	0	0
S3	0.16047	0	0	0	0.83953	0	0	0	0	0	0	0	0	0	0	0
S4	0.12050	0	0	0	0	0.8795	0	0	0	0	0	0	0	0	0	0
S5	0.09764	0	0	0	0	0	0.90236	0	0	0	0	0	0	0	0	0
S6	0.06806	0	0	0	0	0	0	0.93194	0	0	0	0	0	0	0	0
S7	0.06742	0	0	0	0	0	0	0	0.93258	0	0	0	0	0	0	0
S8	0.07631	0	0	0	0	0	0	0	0	0.92369	0	0	0	0	0	0
S9	0.05438	0	0	0	0	0	0	0	0	0	0.94565	0	0	0	0	0
S10	0.08046	0	0	0	0	0	0	0	0	0	0	0.91954	0	0	0	0
S11	0.0625	0	0	0	0	0	0	0	0	0	0	0	0.9375	0	0	0
S12	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0.9	0	0
S13	0.15407	0	0	0	0	0	0	0	0	0	0	0	0	0	0.84593	0
S14	0.22067	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.77933
S15	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Equation 6 is the Markov model. The elements of which can be obtained by substituting in the state transition matrix shown in Table 6.

To check the validity of the model we use the model equation to generate again a time series of 100,000 points. The result is shown in Fig.5. The distribution of periods from the regenerated data using the Markov model is also shown in Table 7 and Fig 6.

Table7: Count of the three successive values generated from model

Period	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Count	1090	496	298	188	137	80	66	88	49	78	44	76	106	122	433

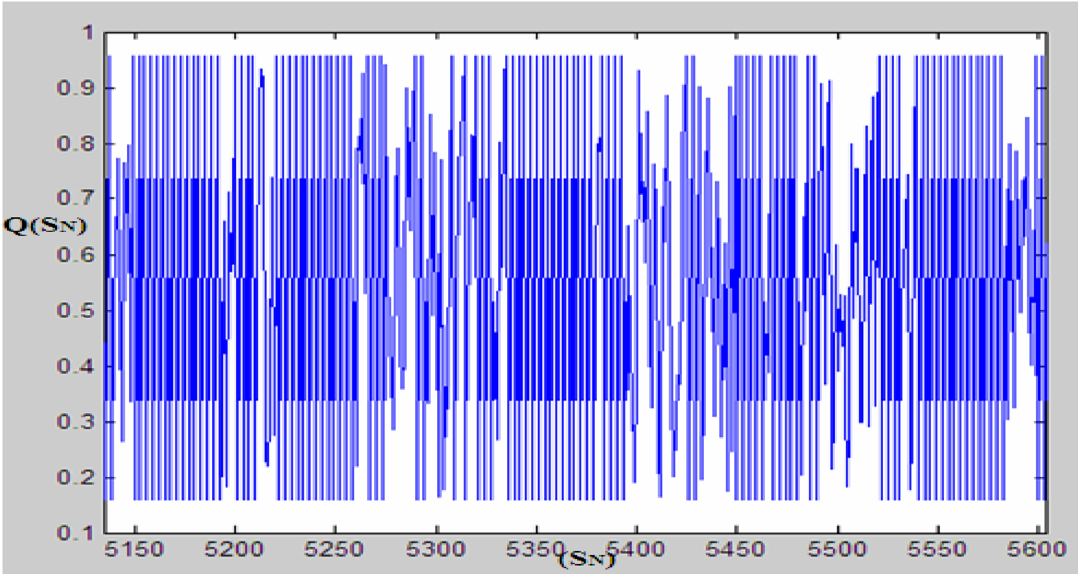

Figure5 The regenerated time series from the Markov model.

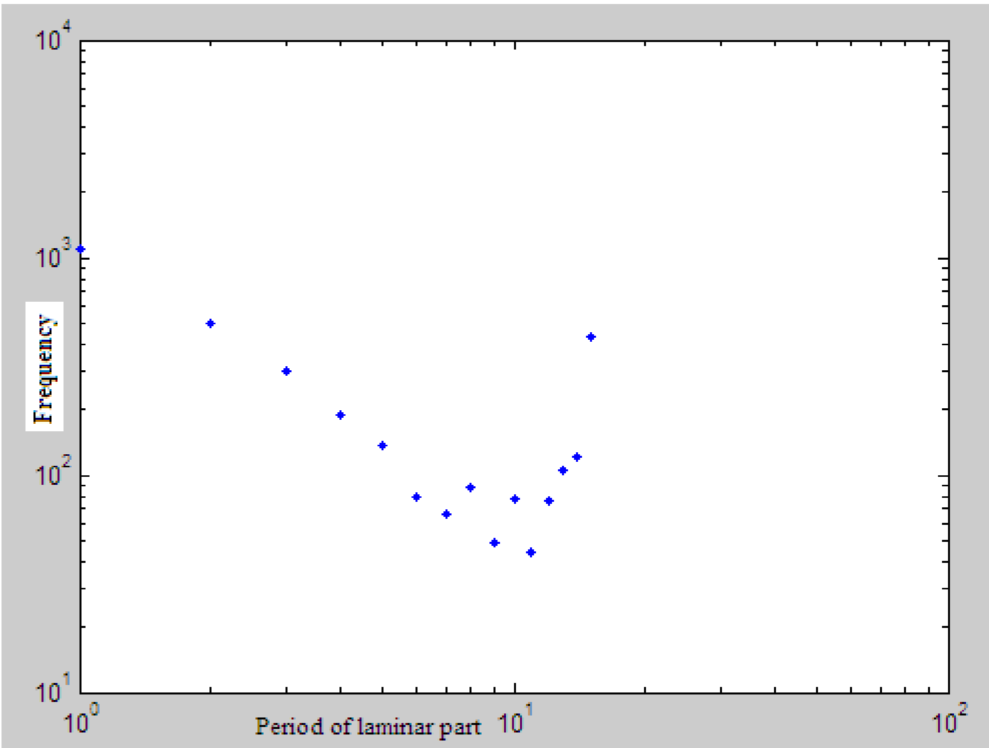

Figure 6 Distribution of periodic part generated from the model.

The model can further be used in reducing or removing the noise or bursting phase and increasing the weight of the laminar phase corresponding to definite mental tasks. This can be easily achieved by tuning the state transition matrix. Tuning may be by increasing the transition probability of one or more periodic state to the next periodic state. Figures 7, and 8 show the results of tuning the state transition matrix.

Now we able to over come the existence a large amount of undesired -chaotic- by determine one or more parameter can control in the existence ratio of chaotic states. We succeed in decreasing the chaotic ratio to more than 65% .

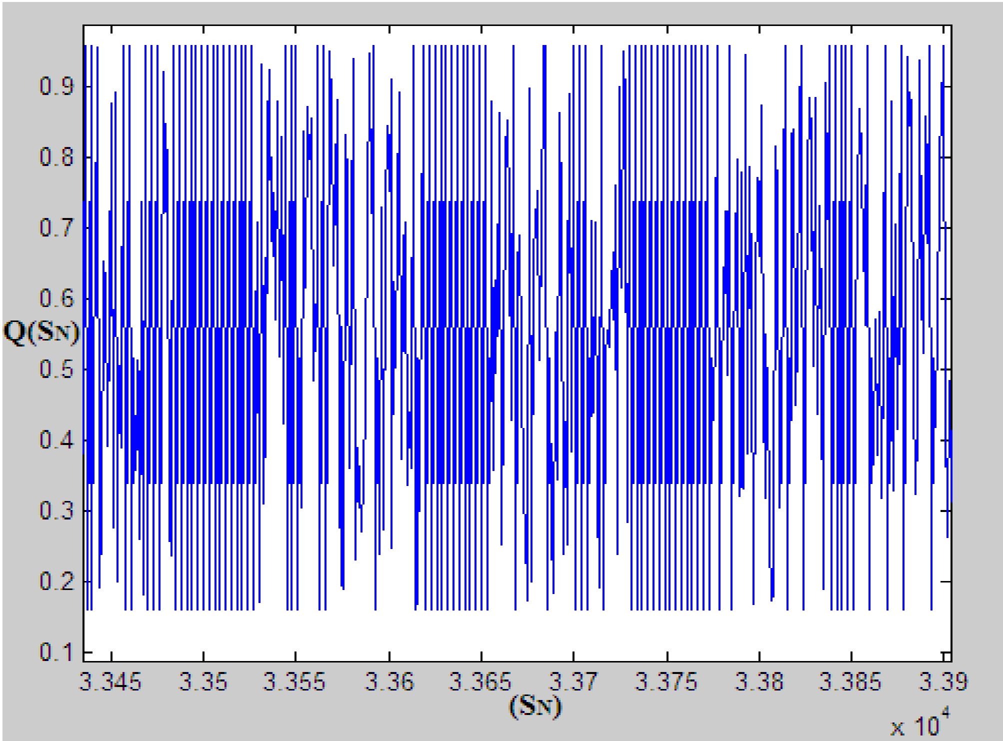

Figure 7 the regenerated time series before tuning (P01=0.22363)

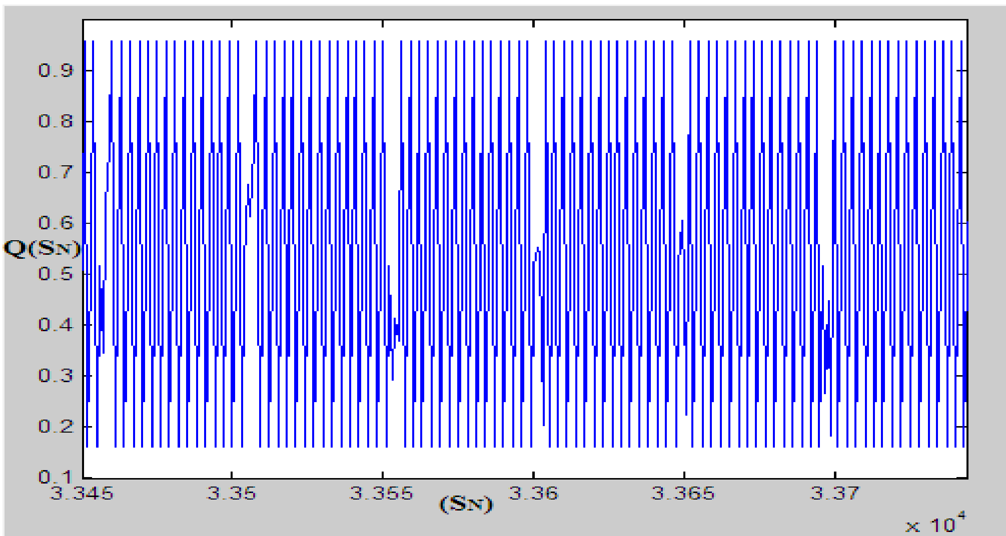

Figure 8 The regenerated time series after tuning the state transition matrix (P01=1)

4. CONCLUSIONS

Due to the difficulty in obtaining real data corresponding to definite mental tasks, a distributed chaotic generator is used to generate a chaotic time series including laminar and burst phases (intermittency) that resemble defined and undefined mental tasks. A Markov model based upon the period count is designed. The model adequacy is tested by regenerating the chaotic time series. The model also shows efficiency in reducing the chaos irregularity.

REFERENCES

1- Cuntai Guan and Fabien Lotte, " Regularizing Common Spatial Patterns to Improve BCI Designs: Unified Theory and New Algorithms published in ", IEEE Transactions on Biomedical Engineering 58, 2 (2011) ", PP355-362, 2011.
2- Nicolas Brodu, Fabien Lotte, and Anatole Lécuyer " Exploring Two Novel Features for EEG-based Brain-Computer Interfaces: Multifractal Cumulants and Predictive Complexity ", published in "Neurocomput-ing 79, 1 (2012) 87-94", PP1-13, 20th May 2011.
3- Fabien Lotte Anatole Lécuyer, and Bruno Arnaldi " FuRIA: An Inverse Solution based Feature Extraction Algorithm using Fuzzy Set Theory for Brain-Computer Interfaces ", published in "IEEE Transactions on Signal Processing", PP1-11, 16th mar 2009.
4- E. A. Hosney, A. A. A. Nasser, and M. I. Sobhy, " Chaotic Behaviour in Nonlinear Circuits", Proceedings of the Eighth National Radio Science Conference, Feb 19-21, 1991, PP1-10/C8, Military Technical College, Cairo, Egypt.
5- Yoko Uwate, Yoshifumi Nishio and Akio Ushida," Markov Chain Modeling of Intermittency Chaos and its Application to Hopfield NN"_,Special Section on Selected Papers from the 16th Workshop on Circuits and Systems in Karuizawa, PP774-779, April2004.

المواجه البينى للمخ والحاسب المبنى على سلاسل ماركوف

يقدم المواجه البينى للمخ والحاسب قناة إتصال مباشرة من المخ، فيعالج نشاط المخ ويترجمه إلى أوامر مستخدما طريقة إستخراج السمه المميزة وخوارزميات التصنيف.

أنظمة المواجه البينى للمخ والحاسب تهدف إلى مساعدة الأشخاص الذين يعانون من الإعاقة الحركية بتزويدهم بالقدرة على التحكم فى أجهزة مثل الكرسى المتحرك والحاسب بالإعتماد على نشاط المخ فقط و دون الحاجة إلى إستخدام الأعضاء العضلية فى الجسم.

نظام المواجه البينى للمخ والحاسب يحدد وجود نمازج محددة فى نشاط المخ لأحد الأشخاص والتى تتعلق بنوايا هذا الشخص ليبدأ التحكم.

أنظمة معالجة الإشارات هى جزء هام جدا فى تصميم أجهزة المواجه البينى للمخ والحاسب لأننا نحتاج إليها فى إستخراج المعلومات التى لها معنى من إشارات المخ.

وكأى نظام إتصال، يوجد للمواجه البينى للمخ والحاسب إدخالات (الإشارات الكهربية الفسيولوجية الناتجة من نشاط المخ) ومخرجات (الأفعال المنفذة من الجهاز).

يقوم الأفراد بالتحكم فى الأجهزة عن طريق الأنشطة العقلية والتى تكون مصاحبة للأفعال إعتمادا على تطبيق المواجه البينى للمخ والحاسب، ويتطلب الإرتباط بين الأنشطة العقلية و الأفعال إختيار مجموعه من الأنشطة العقلية لتشغيل المواجه البينى للمخ والحاسب لتعريف التوقيعات فى نشاط المخ والتى تميز كل نشاط عقلى.

يُنتج نشاط المخ نطاق واسع من الظواهر والتى يمكن قياسها بحساسات مناسبة ولها جهد كهربى يستخدم فى المواجه البينى للمخ والحاسب .

على الرغم من أن نماذج موجات المخ المتعرف عليها محدودة وتمثل عدد محدود من الحالات إلا أن الظروف النفسية تجعله غير محدود. ولتحسين الإستجابه لأمر معين ينبغى أن تزداد هذه التغيرات بحيث يستجيب المخ لأقرب أمر يمثل معنى.

وكان العمل فى هذا البحث لتحسين إستجابة المواجه البينى للمخ والحاسب فى تمييز إشارات المخ -التى تمثل الأوامر التى نحتاج إليها- من الإشارات الأخرى -الضوضاء- والتى تحدث نتيجة تغير الحالة النفسية. ويستخدم المُوَلد العشوائى المُوَزِع لمحاكاة السلوك المتقطع الذى له مرحلة متكررة والتى تمثل المهام العقلية المحددة فى المخ ومرحلة النبضات العشوائية التى تمثل الضوضاء أو المهام العقلية الغير محددة. وأقرب مرحة متكررة فى القسم المتقطع للإستجابة هى المرحلة الثلاثية، وقد تم تطوير الحل الحسابى المبنى على سلسلة ماركوف المخفية بحيث يحتفظ بالإستجابات المتكررة بعيدا عن العشوائية.

وقد نجحنا فى هذا البحث فى توليد متواليه تشبه فى سلوكها إشارات المخ من حيث احتوائها على مجموعات متكررة تحاكى المتولدة من المخ والتى تمثل الأوامر وأخرى عشوائية تشابه الإشارات التى لا تمثل لنا شيئا ولا نحتاجها، وعن طريق إستخدام نموزج سلاسل ماركوف إستطعنا تحديد المجوعات المتكررة والمجموعات العشوائية ووضعها فى صورة حالات، هذه

الحالات تمثل الأنشطة العقلية للمخ، ثم قمنا بحساب إحتمالات التنقل بين هذه الحالات، ومن هذه الإحتمالات قمنا ببناء نظام جديد له ذات النسب تقريبا لتواجد الحالات فى النظام القديم.

وقد قمنا بعدة محاولات لتقليل حجم المجموعات العشوائية وذلك عن طريق زيادة نسبة إحتمالية الإنتقال بين المجموعات المتكررة. وبعد عدة محاولات وجدنا أنه يمكن تقليل المجموعات العشوائية بشكل كبير يصل إلى أكثر من 65% ، مما يساعد على تحسين أداء المواجه البينى للمخ والحاسب.