2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Applications of Probabilistic Model Based on JoyStick Probability Selector

Marko V. Jankovic, *Senior Member, IEEE,* and Nikola Georgijevic

***Abstract*— Recently, it has been shown that a probabilistic model based on two of the main concepts in quantum physics – a density matrix and the Born rule, can be suitable for the modeling of learning algorithms in biologically plausible artificial neural networks framework. It has been shown that the proposed probabilistic interpretation is suitable for modeling on-line learning algorithms for Independent /Principal/Minor Component Analysis, which could be realized on parallel hardware based on very simple computational units. Also, it has been shown that the quantum entropy of the system, related to that model, can be successfully used in the problems like change point detection. Here, it will be shown that the proposed model can be successfully used in other areas of applied signal processing, with some examples of applications in the area of power electronics and general classification problems.**

I. Introduction

It is well known that the field of pattern recognition is concerned with the automatic discovery of regularities in data through the use of computer algorithms and with the use of these regularities to take actions such as classifying the data into different categories. There are many algorithms based on different theoretical backgrounds that could be used for pattern recognition in practical applications (see e.g. [1]). Generally, most of the algorithms are applied in areas like classification, regression or change point detection.

In this paper we are going to present applications based on the recently proposed probabilistic model which was inspired by two of the main concepts in quantum physics (presented here in Sections II and III). It has been shown that proposed probabilistic interpretation is suitable for modeling of on-line learning algorithms for Principal/Minor/Independent Component Analysis, which could be realized on parallel hardware based on very simple computational units [2-4]. In such applications, the proposed concept (model) can be used in the context of improving algorithm convergence speed, learning factor choice, input signal scale robustness, and can be easily deployed on parallel hardware. In section IV.C it will be shown how we create the robust on-line algorithm for single PCA calculation (here it is called BACH algorithm). Also, we are going to show how this probabilistic model can be used in problems like change point detection (Section IV.B), which can find wide use in many areas. Here, we demonstrated the usefulness in several power electronics examples, that are basically related to the problem of change point detection. Application in the area of classification is demonstrated in Sections IV.A

M. V. Jankovic is with the Control Department of Electrical Engineering Institute "Nikola Tesla", University of Belgrade, Koste Glavinica 8a, 11000, Belgrade, Serbia (phone: +381-11-3952-047; fax: +381-11-3690-823; e-mail: elmarkoni@ ieent.org).

N. Georgijevic is with the Power System Department of Electrical Engineering Institute "Nikola Tesla", University of Belgrade, Koste Glavinica 8a, 11000, Belgrade, Serbia (e-mail: nikola.georgijevic@ieent.org).

II. Quantum Probability Model and Born Rule

In quantum mechanics the transition from a deterministic description to a probabilistic one is done using a simple rule termed the Born rule. This rule states that the probability of an outcome ($\boldsymbol{a}$) given a state ($\boldsymbol{\Psi}$) is the square of their inner product $(\boldsymbol{a}^{\mathrm{T}}\boldsymbol{\Psi})^2$. This section is based on a similar section in [5, 3].

In quantum mechanics the Born rule is usually taken as one of the axioms. However, this rule has well established foundations. Gleason's theorem [6] states that the Born rule is the only consistent probability distribution for a Hilbert space structure. Wooters [7] has shown that by using the Born rule as a probability rule, the natural Euclidean metrics on a Hilbert space coincides with a natural notion of a statistical distance. Short review for some other justifications of the Born rule can be seen in [5].

The quantum probability model takes place in a Hilbert space H of finite or infinite dimension. A state is represented by a positive semidefinite linear mapping (a matrix $\boldsymbol{\rho}$) from this space to itself, with a trace of 1, i.e. $\forall\, \boldsymbol{\Psi} \in \mathrm{H}\ \boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\rho}\boldsymbol{\Psi}\geq 0$, $\mathrm{Tr}(\boldsymbol{\rho})=1$. Such $\boldsymbol{\rho}$ is self adjoint and is called a density matrix.

Since $\boldsymbol{\rho}$ is self adjoint, its eigenvectors $\boldsymbol{\Phi}_i$ are orthonormal and since it is positive semidefinite its eigenvalues p_i are real and nonnegative $p_i \geq 0$. The trace of a matrix is equal to the sum of its eigenvalues, therefore $\sum_i p_i = 1$.

The equality $\boldsymbol{\rho}=\sum_i p_i\, \boldsymbol{\Phi}_i \boldsymbol{\Phi}_i^{\mathrm{T}}$ is interpreted as "the system is in state $\boldsymbol{\Phi}_i$ with probability p_i". The state $\boldsymbol{\rho}$ is called the pure state if $\exists i$ s.t. $p_i = 1$. In this case, $\boldsymbol{\rho}=\boldsymbol{\Psi}\boldsymbol{\Psi}^{\mathrm{T}}$ for some normalized state vector $\boldsymbol{\Psi}$, and the system is said to be in state $\boldsymbol{\Psi}$.

A measurement M with an outcome $\boldsymbol{x}$ in some set $\boldsymbol{X}$ is represented by a collection of positive definite matrices $\{\boldsymbol{m}_z\}_{z\in \boldsymbol{Z}}$ such that $\sum_{z\in \boldsymbol{Z}} \boldsymbol{m}_z = \mathbf{1}$ ($\mathbf{1}$ being the identity matrix in H). Applying measurement M to state $\boldsymbol{\rho}$ produces an outcome $\boldsymbol{x}$ with probability

$$p_x(\boldsymbol{\rho})=\mathrm{trace}(\boldsymbol{\rho}\boldsymbol{m}_x). \tag{1}$$

This is the Born rule. Most quantum models deal with a more restrictive type of measurement called the von Neumann measurement, which involves a set of projection operators $\boldsymbol{m}_a=\boldsymbol{a}\boldsymbol{a}^{\mathrm{T}}$ for which $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{a}'=\delta_{\boldsymbol{a}\boldsymbol{a}'}$. In a modern language, von Neumann's measurement is a conditional expectation onto a maximal Abelian subalgebra of the algebra of all bounded operators acting on the given Hilbert space. As before, $\sum_{\boldsymbol{a}\in \mathrm{M}}\ \boldsymbol{a}\ \boldsymbol{a}^{\mathrm{T}} = \mathbf{1}$. For this type of measurement the Born rule takes a simpler form:

$p_a(\boldsymbol{\rho})=\boldsymbol{a}^T\boldsymbol{\rho}\boldsymbol{a}$. Assuming $\boldsymbol{\rho}$ is a pure state this can be simplified further to

$$p_a(\boldsymbol{\rho}) = (\boldsymbol{a}^T\boldsymbol{\Psi})^2. \quad (2)$$

So, we can see that the probability of the outcome of the measurement will be $\boldsymbol{a}$, if the state is $\boldsymbol{\rho}$, is actually the cosine square of the angle between vectors $\boldsymbol{a}$ and $\boldsymbol{\Psi}$, or $p_a(\boldsymbol{\rho})=\cos^2(\boldsymbol{a},\boldsymbol{\Psi})$.

III. JoyStick Probability Selector

In this section we will give the recently proposed, simple interpretation of the probability that is related to the Born rule [3]. Here we will assume that we are dealing with finite dimensional discrete variable.

For the moment, let's assume that we are dealing with discrete two dimensional variables. It can associate us with a coin tossing. Assume further that the two possible outcomes of our experiment are represented by the dummy variables {01} and {10}. If we represent our coin as a unit norm vector in the two dimensional space (we will call that vector JoyStick Probability Selector or JSPS), then we can have the following simple geometric interpretation given in Fig. 1.

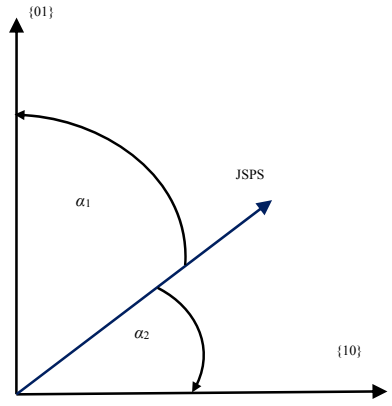

Fig. 1. JoyStick Probability Selector

Now, we will suggest that the probability of outcome {01} is equal to the cosine square of angle α_1, while the probability of outcome {10} is equal to the cosine square of angle α_2. It is not difficult to see that $\cos^2(\alpha_1) + \cos^2(\alpha_2) = 1$. We can see that the probability of the particular outcome of the experiment (in this case coin toss) is equal to the inner product of the unit norm JSPS and the unit norm vector which represents that outcome. Then, it is easy to see that this coincides with the Born rule interpretation for the case of a pure state and von Neumann measurement system. It is easy to see that when state (JSPS) vector collapses to one of the states, it is not possible to give information about the other state, which is consistent with some quantum mechanics results.

We can check what will happen if our discrete variable is of the dimension 3. In that case our system can be represented by Fig. 2. Now, we have three possible outcomes of the experiment that are represented by dummy variables {001}, {010} and {100}. Again we have the JSPS vector which represents the status of our variable before we perform the measurement. Again, the probability of the outcome is given by the cosine square of the angle between JSPS and particular outcome vector. It is not difficult to check that

$$\sum_{i=1}^{3}\cos^2(\alpha_i)=1. \quad (3)$$

This follows from the Pythagorean Theorem. For any 3-D vector whose norm is r we have

$$\sum_{i=1}^{3} r^2\cos^2(\alpha_i)=r^2. \quad (4)$$

It can be easily concluded that this way of reasoning can be extended to any finite dimension D.

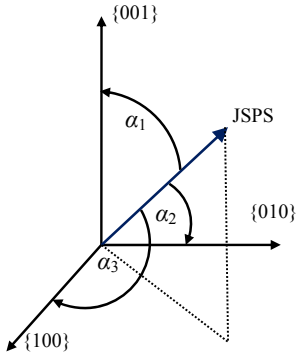

Fig. 2. A 3-D example

It is interesting to notice that this representation is actually reasonable for unimodal data (data that can be represented by a pure state). If the data is multi modal we need more complex model which is not going to be analyzed here.

IV. Quantum Probability Model in Neural Networks Context

In this section, we recapitulate some quantum probabilistic concepts that can be used in a neural network framework. We show how neural networks can be used in a probabilistic framework that is basically based on the Born rule.

The basic single layer feedforward artificial neural network is depicted in Fig. 3. The output of the n-th output unit y_n (n=1, 2,…, N) of a layer of parallel linear artificial neurons is given as

$$y_n(i)=\boldsymbol{w}_n(i)^T\boldsymbol{x}(i), \quad (5)$$

with $\boldsymbol{x}(i)$ denoting a K-dimensional zero-mean input vector of the network and $\boldsymbol{w}_n(i)$ denoting a weight vector of the n-th output unit, and i representing sampling instances iT, where T is a sampling period. The output vector $\boldsymbol{y}$ is defined as

$$\boldsymbol{y}(i)=\boldsymbol{W}(i)^T\boldsymbol{x}(i). \quad (6)$$

In the usual interpretation, based on specific requirements, e.g. minimization of some cost function, matrix $\boldsymbol{W}$ is changed (trained) in the process of learning, according to some adopted learning rule.

Here we will give a slightly different interpretation. We will consider a Hilbert space H of a finite dimension. "State vectors" are defined by the input data vector $\boldsymbol{x}_k$ and we can imagine that every vector $\boldsymbol{x}_k$ is available in a big enough number of copies (clones), so that we can perform as many simultaneous measurements as we want. A measurement M with an outcome $\boldsymbol{w}_n$ in some set $\boldsymbol{W}$ is represented by a collection of positive definite matrices $\{\boldsymbol{m}_{w_n}\}_{w_n \in \boldsymbol{W}}$ such that $\boldsymbol{m}_{w_n} = \boldsymbol{w}_n \boldsymbol{w}_n^{\mathrm{T}}$, so $\Sigma_{w_n \in \boldsymbol{W}} \boldsymbol{m}_{w_n} = \boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}$, which is not necessarily equal to the identity matrix on H. This means that the sum of the probabilities of the particular outcomes does not have to be equal to one – in other words, sometimes we will work with improper discrete distributions. Also, measures like entropy and divergence will be applied to improper probability distributions, or to a mixture of proper and improper probability distributions. In the following sections, we will point out that in the adopted framework, this will not affect the final result.

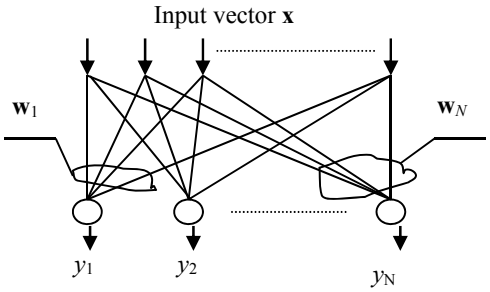

Fig. 3. A layer of parallel linear artificial neurons

Applying measurement M to state $\boldsymbol{x}_k$ produces the outcome $\boldsymbol{w}_n$ with the probability (the Born rule)

$$p(\boldsymbol{w}_n \mid \boldsymbol{x}_k) \stackrel{def}{=} \cos^2(\boldsymbol{w}_n, \boldsymbol{x}_k), \tag{7}$$

regardless of the norm of the vectors $\boldsymbol{w}_n$ and $\boldsymbol{x}_k$. In the following text, we will consider only vectors $\boldsymbol{w}_n$ that have unit norms. This means

$$p(\boldsymbol{w}_n | \boldsymbol{x}_k) = \frac{\left(\boldsymbol{w}_n^{\mathrm{T}} \boldsymbol{x}_k\right)^2}{\|\boldsymbol{x}_k\|^2}. \tag{8}$$

Also, if we apply N simultaneous measurements M^N to the state $\boldsymbol{x}_k$ we obtain outcome $\boldsymbol{W}$ with the probability

$$p(\boldsymbol{W} \mid \boldsymbol{x}_k) \stackrel{def}{=} \sum_{n=1}^{N} p(\boldsymbol{w}_n \mid \boldsymbol{x}_k). \tag{9}$$

Here, it is assumed that the outcome of each measurement could be different. We define the joint probability of the state $\boldsymbol{x}_k$ and outcome $\boldsymbol{W}$ obtained by simultaneous multiple measurement M^N on state $\boldsymbol{x}_k$, $p(\boldsymbol{W}, \boldsymbol{x}_k)$ as

$$p(\boldsymbol{W}, \boldsymbol{x}_k) \stackrel{def}{=} p(\boldsymbol{W} \mid \boldsymbol{x}_k) p(\boldsymbol{x}_k). \tag{10}$$

Now, without loss of generality, let's assume that we are dealing with a random variable $\boldsymbol{x}$ that takes realizations from a set of observed K-dimensional zero-mean data vectors $\{\boldsymbol{x}_k\}$, $k \in \{1, \ldots, N_{\mathrm{sample}}\}$, which are sampled from some distribution in time instants $t = kT$ where k is already defined and T represents the sampling period. Then, we can define $p(\boldsymbol{x}=\boldsymbol{x}_k \mid t=kT)$ as

$$p(\boldsymbol{x}_k) \stackrel{def}{=} \frac{\|\boldsymbol{x}_k\|^2}{\sum_{i=1}^{N_{sample}} \|\boldsymbol{x}_i\|^2}, \tag{11}$$

where N_{sample} represents the overall number of samples that are going to be analyzed. It is interesting to note that the only thing that we can conclude about the $p(\boldsymbol{x}_k)$ is that it is proportional to $\|\boldsymbol{x}_k\|^2$. The sum in the denominator represents the energy of samples that are going to be analyzed – we actually do not know the value of that sum at any, but the final moment. However, we know that it represents some constant. We can easily see that the adopted probability measure fulfils the two conditions that are required for the probability function $f(z)$ (in our case $p(z)$) to be considered as a modified generalized probability measure [8]:

1. For each z, $0 \le f(z) \le 1$,
2. $\sum_i f(z_i) = 1$.

In this definition, orthonormality is not explicitly required in order that the coefficients $f(z_i)$ sum up to one. However, from the JSPS introduction [3], we can see that it is always implicitly present.

Here, we will consider all vectors as "oriented energies" or

$$\boldsymbol{x}_k = \|\boldsymbol{x}_k\| \frac{\boldsymbol{x}_k}{\|\boldsymbol{x}_k\|} = \|\boldsymbol{x}_k\| \boldsymbol{x}'_k, \tag{12}$$

where the norm of the vector $\|\boldsymbol{x}_k\|$, represents the square root of the energy contained in the vector $\boldsymbol{x}_k$, and the orientation represents some unit norm vector $\boldsymbol{x}'_k$, which represents some pure state. In that case, we can see that the statistical description of our system is represented by the density matrix $\boldsymbol{\rho}$

$$\boldsymbol{\rho} = \sum_k p_k \boldsymbol{x}'_k \boldsymbol{x}'^{\mathrm{T}}_k, \tag{13}$$

as a statistical mixture of pure states $\boldsymbol{x}'_k$, and $p_k = p(\boldsymbol{x}_k)$ are defined by (7). We have to stress that the density matrix $\boldsymbol{\rho}$ that is created here, does not fulfill the requirements of quantum mechanical postulates, since it connects the pure states from different time instants. However, we used this term here to stress the conceptual analogy with original definition of density matrix (although we could create a new term – e.g. normalized covariance matrix). We can see that

$$\boldsymbol{\rho} = \frac{N_{sample}}{\sum_{i=1}^{N_{sample}} \|\boldsymbol{x}_i\|^2} \boldsymbol{C}, \tag{14}$$

where $\boldsymbol{C}$ is the input signal covariance matrix. Obviously, the matrix $\boldsymbol{\rho}$ and the matrix $\boldsymbol{C}$ have the same eigenvectors.

In the proposed context, the learning algorithm applied to the neural network has a basic task - to find the measurement system in which input data is "best explained", or have the features that are specified. As an example, principal component analysis will search for the measurement (or we can say coordinate) system in which the input data covariance matrix is diagonal.

V. Practical Implementation

In this section it is shown how the proposed model can be used in various applications.

A. Applications in simple classification problems

Here, it will be shown how this concept can be used to solve some classification problems. It will be shown, how we can solve XOR problem using quantum Tsallis entropy [10]. The Tsallis entropy is defined by the following equation:

$$E_{TS} = \frac{1 - \sum_{k=1}^{K} p(x_k)^q}{q-1}, \tag{15}$$

where $q>0$.

Here we use target value as an input. In other words, we consider the problem as a 3-dimensional problem where the 3rd dimension is defined by output value. Input vectors are defined as

$$\boldsymbol{a}_1 = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}, \quad \boldsymbol{a}_2 = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}, \quad \boldsymbol{b}_1 = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} \text{and} \, \boldsymbol{b}_2 = \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}. \tag{16}$$

Matrix $\boldsymbol{A}$ which describes the system in the plane z=-1 (we assume vectors are presented in 3-dimensional system x-y-z) is defined as $\boldsymbol{A}=\boldsymbol{a}_1 \cdot \boldsymbol{a}_1^{\mathrm{T}} + \boldsymbol{a}_2 \cdot \boldsymbol{a}_2^{\mathrm{T}}$, while the system in plane z=1 is defined as $\boldsymbol{B}=\boldsymbol{b}_1 \cdot \boldsymbol{b}_1^{\mathrm{T}} + \boldsymbol{b}_2 \cdot \boldsymbol{b}_2^{\mathrm{T}}$. Then we can calculate the entropy of the individual systems E_a and E_b. Overall entropy of the unified system is calculated as $E=E_a+E_b$. Then, for any x and y coordinate, we can calculate how it affects the entropy of the overall system, by adding it to system $\boldsymbol{A}$ or system $\boldsymbol{B}$. We will label it in such a way that addition of the individual point to one of the system creates a minimum change in the overall entropy. The result of such calculation is shown in Fig. 4. From the figure we can see that the problem is successfully solved. Other simple problems could be solved by this method, like the AND, OR classification, or the IRIS problem classification. For more complicated problems, we need refinement of the calculation of overall entropy, and it is not going to be addressed here. It is going to be stressed here that in the case of n-dimensional data, regardless of the number of the points that are analyzed, we need to store k (where k is the number of classes) matrices of dimension nxn.

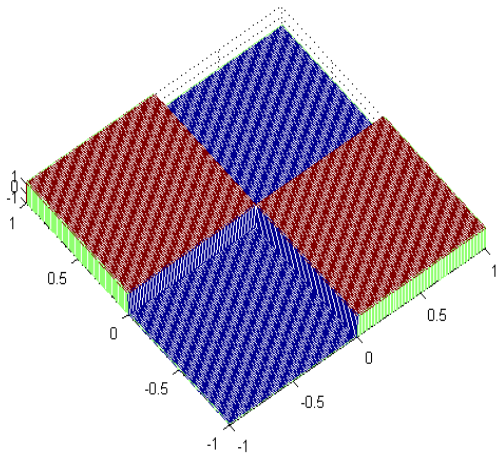

Fig. 4. XOR problem classification

The logical AND problem could be solved in a similar way. However, since we have an unbalanced system it is necessary to define four subclasses and to evaluate the overall entropy for the four possibilities – meaning that any point can potentially belong to any of the four subclasses. The result of classification is shown on Fig. 5.

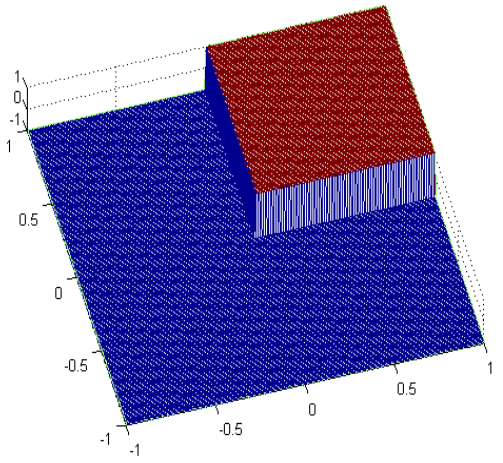

Fig. 5. AND problem classification

Similar method could be used in the famous IRIS classification problem. Taking some points as characteristic representatives (in this case points numbered (15-34) from each group) we can correctly classify the rest of the points. As was the case in the logical AND classification problem, here we have 45 subclasses, and we have to evaluate the entropy for 45 cases – for every point we want to classify, we can assume that it can belong to any of the 45 subclasses. Although it can look time consuming, we actually have 45 simple independent processes that could be easily realized on parallel hardware, like GPU. For illustration we can use the following Fig. 6, where we used only attributes 3 and 4 for classification (we can see that this classification is not 100% correct – it is necessary to add attribute 1 too, but we cannot present it graphically).

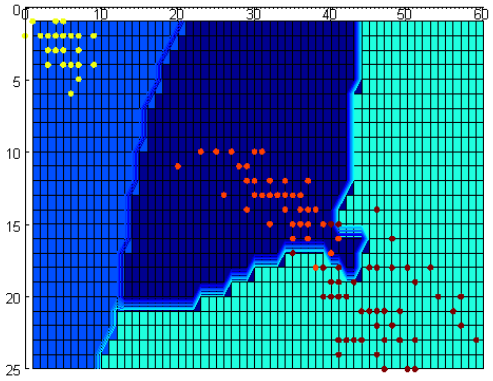

Fig 6. Classification of IRIS data if we use only the data specified by attributes 3 and 4

B. Applications in power electronics

Here we will show that proposed method could be successfully used in several applications in power electronics, like in detection of high voltage induction motor rotor failure, detection of thyristor malfunction in rectifiers or power quality analysis. All problems are related to the problem of change point detection.

1) Detection of high voltage inductor motor rotor failure

Here, we recapitulate an example of rotor failure detection [9]. The problem under scope is how to make a very simple and cheap device for rotor failure detection in large induction motors in an early stage, in order to increase production availability through reduced downtime and by minimizing the repair cost. That simple and cheap device should be connected to each induction motor under surveillance. The main problem is how to detect the first sign of rotor failure during normal operation.

Example of total rotor failure (several cracked and broken rotor bars) on the motor is shown on Fig.7.

Fig. 7. The faulty cage with several cracked and broken rotor bars

In [9] it was proposed that the measure of the "behavior" of the motor current is Tsallis' entropy of the recorded samples in the sliding window of the proper size. In this case, q is usually taken in the range 6-12, and x_k represents samples of the derivatives of the stator current and $p(x_k)$ is defined by equation (3). The measure M of the current "behavior" is actually defined by the following equation

$$M = \mathrm{E}\left(\sum_{k=1}^{K} p(x_k)^q\right) = \mathrm{E}\left(1-(q-1)E_{TS}\right), \qquad (17)$$

where E stands for expectation. In order to improve the method sensitivity, the derivatives of the stator current are used for numerical processing. The reason being that the rotor failure could be detected by the presence of the spectral components close to the 5th and 7th harmonic, which are much more "visible" in the derivative of the stator current than in the stator current itself. On the following Fig. 8, we can see the sinusoidal signal with 1% presence of the 5th harmonic, on the upper figure and corresponding the derivative shown on the lower figure. By calculating the factor M for the case without presence of the component close to 5th harmonic and with its 1% presence, change of the M of the stator current is just 20%, while the change of M of the derivative is close to 300% (for q=6).

In the case that it is necessary to target some specific frequency "region", it is possible to condition a signal with an appropriate band-pass filter and, again it is possible to use M as a measure of the change in "behavior".

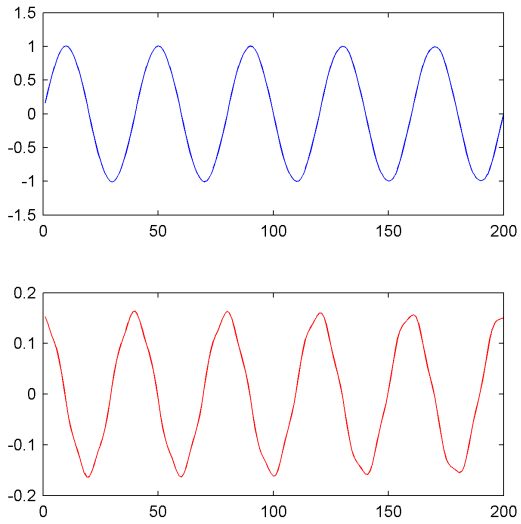

Fig. 8. Sinusoidal signal with 1% presence of 5th harmonic (upper plot) and its derivative (lower plot)

2) Detection of thyristor malfunction in rectifier bridge

Three-phase controlled rectifiers have a wide range of applications, from small rectifiers to large High Voltage Direct Current (HVDC) transmission systems. They are used for electrochemical processes, many types of motor drives, large power generators excitation systems, traction equipment, controlled power supplies, and many other applications. From the point of view of the commutation process, they can be classified in two important categories: Line Commutated Controlled Rectifiers (Thyristor Rectifiers) and Force Commutated PWM Rectifiers [11].

One of the problems related to Thyristor Rectifiers is possible malfunctioning of one of the thyristors while the rest of the bridge is still working and controls the average output value correctly. In that case, the presence of the harmonics in the output voltage is bigger and causes problems to some equipment, while on the other side, it

puts a bigger stress on other working thyristors and can cause the malfunctioning of the whole rectifier. For that reason, it is necessary to detect thyristor malfunction as soon as possible. Here we will show that we can use the Tsallis entropy of the bridge voltage to detect the problematic thyristor. Using the measure M defined in (5) we can easily detect the problem, and if we have some additional information about synchronization, we can correctly identify the thyristor that causes the problem. Some characteristic waveforms are presented on Fig. 9. Corresponding M values are presented on Fig. 10 and Fig. 11.

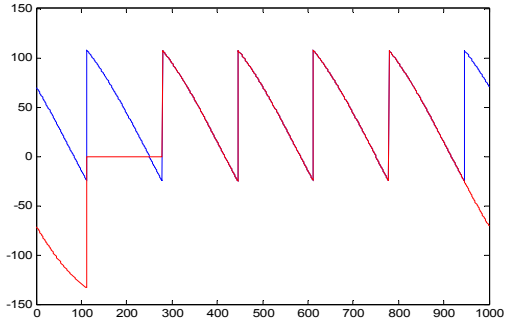

Fig. 9. Characteristic one-period waveforms for the correct bridge (blue line) and the bridge with malfunctioning thyristor (red line)

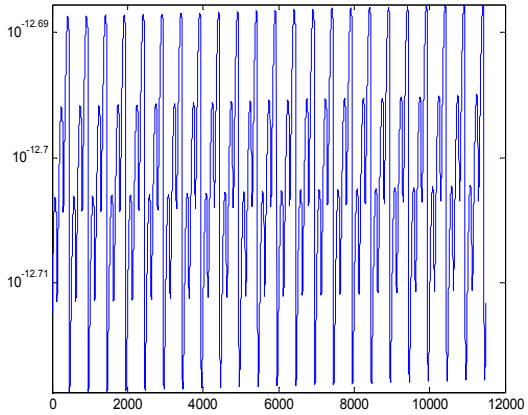

Fig. 10. M values for a correct thyristor bridge

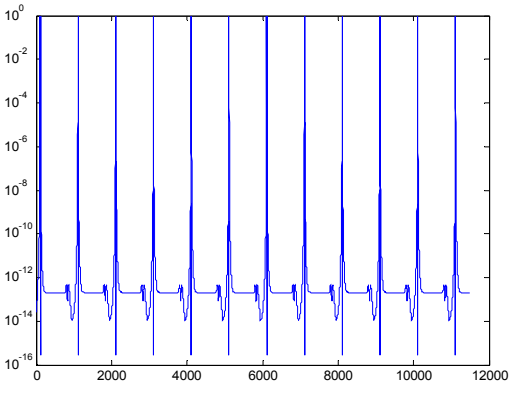

Fig. 11. M value for a bridge with a malfunctioning thyristor

From figures 10 and 11 we can see that the problem can easily be detected and with some additional information we can detect exactly which of the thyristors is problematic.

3) Power quality analysis

The quality of electric power has become an important issue for electric utilities and its customers. As a result, power quality (PQ) study is gaining interest. Degradation in quality of electrical power is normally caused by power-line disturbances such as voltage sag/swell with and without harmonics [12]. Momentary interruption, harmonic distortion, flicker, notch, spike and transients, are causing problems such as malfunctions, instabilities, short lifetime, failure of electrical equipment and so on. In a realistic distribution system, in order to improve power quality, these disturbances need to be identified before appropriate mitigation action can be taken. In this paper we will consider some of the disturbances that occur frequently, like sag, swell, flicker or presence of harmonics. On the following figures (Fig. 12-16) we can see the characteristic waveform of signals and the calculated value M (based on (17)). It is clear that there is a characteristic signature of all disturbances which can be separated by classification algorithms (e.g. similar to method explained in section V.A).

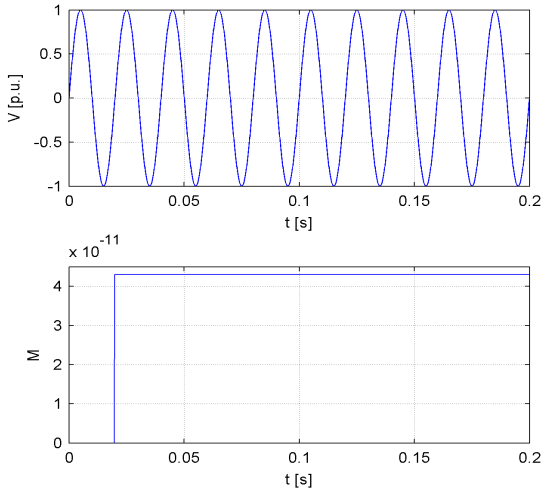

Fig. 12 Pure sinusoidal signal (upper part) and the value M (lower part)

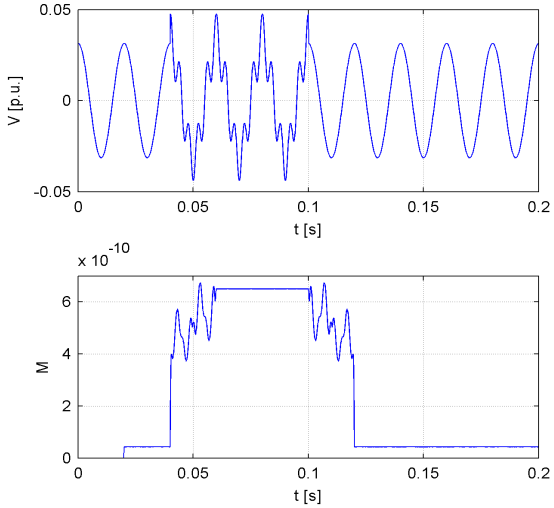

Fig. 13 Sinusoidal signal with harmonics in one part (upper part) and the value M (lower part)

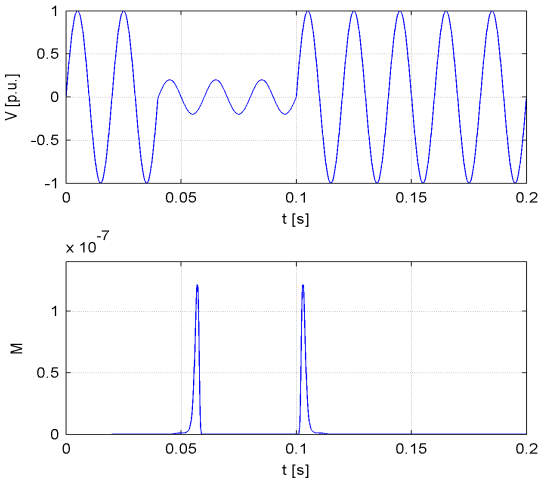

Fig. 14 Sinusoidal signal with sag in a part (upper part) and the value *M* (lower part)

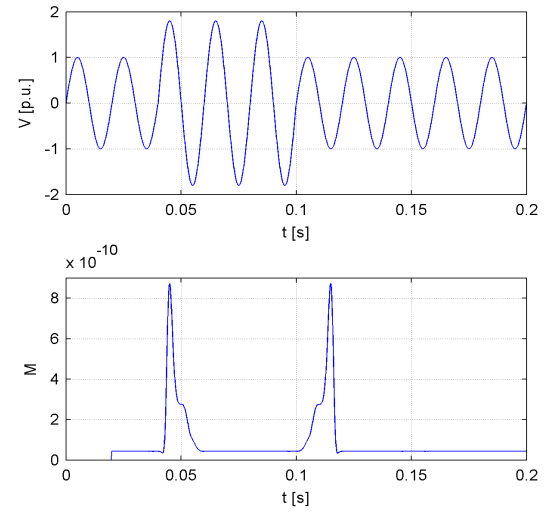

Fig. 15 Sinusoidal signal with a swell (upper part) and the value *M* (lower part)

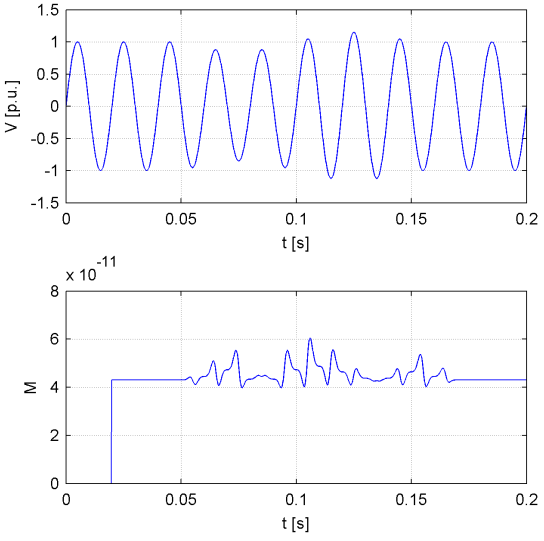

Fig. 16 Sinusoidal signal with flicker (upper part) and the value *M* (lower part)

C. Robust on-line algorithm for PCA calculation

In this section, we will propose a new algorithm for the extraction of the single principal component. We will call it the BACH algorithm. It is based on minimization of the BACH divergence [13]. Bach divergence is defined as

$$D_{BACH} = (P\|Q) = E\left(\left(p^b - q^b\right)^2\right). \tag{18}$$

For illustration, we will assume that we have 16-dimensional input signal $\boldsymbol{x}$ sampled from uniform distribution. Our learning rule is given as

$$\Delta w \approx \frac{\left(xy^T - wy^2\right)\left(\left(x^T x\right)^b - \left(y^T y\right)^b\right)}{\left(y^T y\right)^{1-b}}, \tag{19}$$

where b is a positive real number. Here, we select b=0.025. Without going into details, we can remove term $((x^Tx)^b-(y^Ty)^b)$, so the BACH learning rule can be represented as

$$\Delta w \approx \frac{\left(xy^T - wy^2\right)}{\left(y^T y\right)^{1-b}}. \tag{20}$$

We can see that it represents modification of the famous Oja learning rule [14], which is, as we are going to see from the illustration example, robust to change of the input signal energy level, which significantly simplifies selection of the learning rate. In the Fig. 17, we can see the behavior of the BACH (solid line) and Oja's algorithm for the "nominal" energy level.

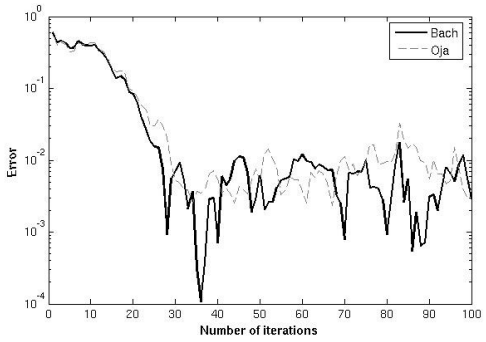

Fig. 17 Experimental results for BACH PCA algorithm (16-dimensional input, 4-dimensional output)

We can see that both algorithms behave similarly. However, if we decrease the input energy 100 times, and keep learning rates unchanged, we can see (Fig. 18) that convergence speed of the BACH algorithm is only slightly affected by the input signal energy change, while Oja's algorithm becomes very slow. Although it is not going to be presented here, it can be said that the BACH algorithm will not be significantly affected, even in the case when the input energy is decreased by several order of magnitude (e.g. billion times).

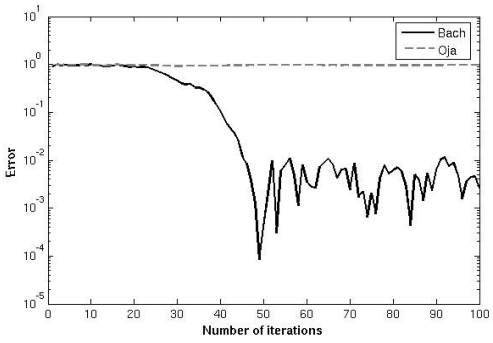

Fig. 18. Experimental results for BACH PCA algorithm (input signal energy decreased 100 times)

VI. CONCLUSION

In this paper we used the recently proposed probabilistic model based on the Born rule and its simple geometrical representation, named JoyStick Probability Selector. It has been shown that the proposed model could be useful in several areas of applied signal processing. It has been shown that the proposed model accompanied with related entropy functions could be used for a generation of simple classification methods that are not computationally intensive. Also, on several examples from the area of power electronics, it has been shown that the model is useful for contraction of the methods for change point detection. It is demonstrated how the proposed model could be used for development of a robust on-line learning algorithm for single PCA computation.

It should be noticed that application to general classification problems still requires creation of the method for class representative points selection (which is analogous to support vectors). This is currently under investigation.

REFERENCES

[1] C.M. Bishop, *Pattern Recognition and Machine Learning.* New York: Springer Science+Business Media, LLC, 2006.

[2] M. Jankovic, M. Manic, and B. Reljin, "Probabilistic model for minor component analysis based on Born rule", In Proceedings of *11th Symposium on Neural Network Applications in Electrical Engineering* (NEUREL 2012), pages 85-88, Belgrade (Serbia), 2012

[3] M. Jankovic and M. Sugiyama, "Probabilistic principal component analysis based on JoyStick Probability Selector", In Proceedings of *International Joint Conference on Neural Networks* (IJCNN 2009), pages 1414 – 1421, Atlanta (USA) , 2009

[4] M. Jankovic and N. Rubens, "A new probabilistic approach to independent component analysis suitable for on-line learning in artificial neural networks", *In Proceedings of the ICONIP 2012*, Part III, Lecture Notes in Computer Science 7665, pages. 552-559, Springer-Verlag , 2012.

[5] L. Wolf, "Learning using the Born rule", Tech. Rep. MIT-CSAIL-TR-2006-036, 2006.

[6] A. M. Gleason, "Measure on the closed subspaces of a Hilbert space", *J. Math. & Mechanics,* vol. 6, pp. 885-893, 1957.

[7] W. K. Wooters, "Statistical distance and Hilbert space", *Physical Review D.*, vol. 23, pp. 357-362, 1981.

[8] M. K. Warmuth and D. Kuzmin, "A Bayesian probability calculus for density matrices", in *Proc. 22nd Conference on Uncertainty in Artificial Intelligence*, pp. 503-511, 2006.

[9] M. Jankovic, T. Gajic and Z. Janda, "Simple method for rotor failure detection based on quantum entropy", 17th International Symposium on Power Electronics Ee-2013, T.3.6 , Serbia, 2013.

[10] C. Tsallis, "Possible generalization of Boltzmann-Gibbs statistics", *J. Statistical Physics,* vol. 52, pp. 479-478, 1988.

[11] S. Mishra, C.N. Bhende and B.K. Panigrahi, "Detection and classification of power quality disturbances using S-transform and probabilistic neural networks", *IEEE Trans. on Power Delivery,* vol. 23, pp. 280-287, 2008.

[12] M. Rashid, *Power Electronics Handbook*, Butterworth-Heinemann, USA, 2011.

[13] M. Jankovic, "Probabilistic approach to neural networks computation based on quantum probability model – probabilistic principal subspace analysis example", Tech. Rep. arXiv:1001.4301v1, 2010.

[14] E. Oja, "A Simplified neuron model as a principal component analyser", J. Math. Biol., vol. 15, pp. 267-273, 1982