FPGA and Non-extensive particle filter for the drone autonomous navigation

Jose Renato G. Braga[a], Elcio H. Shiguemori[b], Haroldo F. de Campos Velho[a 1], Patrick Doerthy[c]

[a]National Institute for Space Administration (INPE), São José dos Campos, Brazil

[b]Department of Aeronautics Science and Technology (DCTA), São José dos Campos, Brazil

[c]Linköping University, Linköping, Sweden

Abstract

A strategy employing image processing is applied to estimate the drone position for autonomous navigation: visual odometry, and computer vision. The computer vision approach is based on the image edge extraction, where segmented image correlation is applied for matching objects on reference and drone images. The edge detection step is computed by a neural network implemented on FPGA to speed up the processing. Data fusion combining the two estimation positioning systems is computed by using the non-extensive particle filter (NExt-PF). The NExt-PF is designed with the likelihood operator using the Tsallis' probability distribution. Our approach shows better results than stantard particle filter (Gaussian likelihood).

Keywords: UAV autonomous navigation, image processing, neural network, FPGA, Non-extensive particle filter.

1. Introduction

Nowadays, the Unmanned Aerial Vehicles (UAV) is a technology with many applications. With the increase of UAV employment, the development of autonomous navigation systems has been a relevant research topic, where the estimation of the UAV position is an important issue. Combining signals from an Inertial Navigation System (INS) with a Global Navigation Satellite System (GNSS) can be used as a positioning drone method. One alternative is to use image processing procedures for the UAV location: visual odometry and computer vision.

Our approach uses images from visible band drone camera and satellite images. Here, the data fusion is performed by a non-extensive particle filter

[1]E-mail Corresponding Author: haroldo.camposvelho@inpe.br

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

(NExt-PF). For the validation of the proposed method, a simulation using images from experiments in the Linköping University (Sweden). The approach results are promising for the UAV position estimation. The method also can be employed for uncertainty quantification

2. Autonomous Navigation by Image Processing

2.1 Visual odometry for UAV positioning

Visual odometry (VO) is a technique to estimates the vehicle position and orientation by processing the changes in the images caused by its movement [11]. Here, the monocular VO is applied to estimate the movement of outdoor UAVs. The VO basic principle is detecting interest points in the image and extract a data structure. From the interest points matching, it is possible to estimate the vehicle motion.

The visual odometry is carried out by four steps. The operation *Image Sequence* uses two images captured at two instants $t - \Delta t$ and t. The value of Δt must be appropriated to display the most part of the scene for both images. *Detecting Points* operation finds the interest points in both images, by using the the *Speeded Up Robust Features* (SURF) [2]. The last step is the *Motion* estimation, for determining the moviment of the UAV with pairs of corresponding interest points. The eight-point algorithm is applied for motion estimation. This algorithm can be described by calculating the fundamental matrix F:

$$(\mathbf{x}')\, F\, \mathbf{x} = 0 \ . \tag{1}$$

There is just one matrix $\mathbf{F}$ satisfying Eq. (1). The eight-point algorithm finds the fundamental matrix, and using the Singular Value Decomposition (SVD) is possible to get the vehicle motion, represented by the rotation matrix and the translation vector:

$$\mathrm{SVD}\{\mathbf{F}\} = \mathbf{K}^T\, \mathbf{R}\, [t]_x\, \mathbf{K}^{-1} \tag{2}$$

where the $\mathbf{K}$ is the matrix of the intrinsic sensor parameters linked to the vehicle, $\mathbf{R}$ is the rotation matrix, and $[t]_x$ is the representation of the cross product of the translation vector.

2.2 Computer Vision Approach

The computer vision technique employed here is multi-step process – see Figure 1. For the first step, the UAV and reference images are changed to gray-scale. After that, a median filter is applied to the reference and UAV

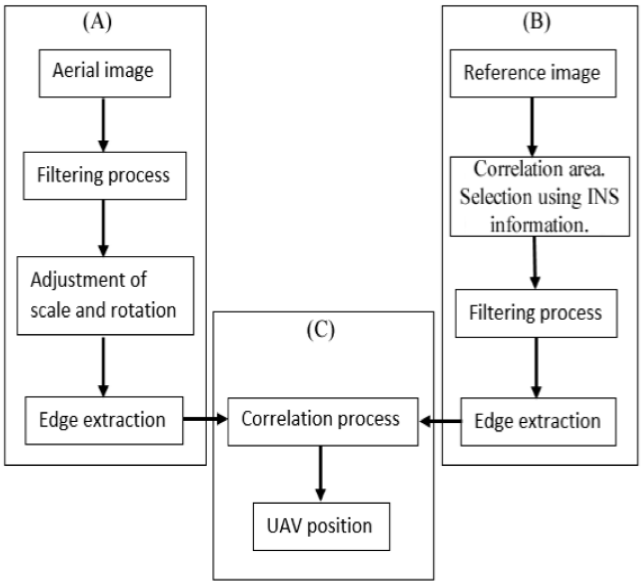

Figure 1: Representation of the computer vision approach for a UAV positioning system.

images to mitigate the noise on the images [8]. Image edge extraction, the process with more computational effort, is performed by a *multi-layer perceptron* (MLP) neural network [9], using *back-propagation* training algorithm. Canny and Sobel approaches for edge identification are also implemented for comparison. The best neural network architecture can be determined by minimizing a functional. The optimization problem is solved by using a new meta-heuristic called Multi-Particle Collision Algorithm (MPCA) [10].

2.2.1 Self-configuring artificial neural network

Artificial neural network is a mathematical model, inspired by human brain processes. The MLP is a supervised feedforward neural network widely used. The architecture of this neural network has three main features: one input layer – non-computational neurons –, one or more hidden layers composed of computational units (neurons), and one output layer – can be computational neurons. The MLP is a NN with neurons fully connected. The connection weights need to be computed by a training algorithm..

The appropriated configuration for a supervised neural network can be

a difficult task. There are many parameter/functions for the MLP-NN to be determined. The MPCA is a metaheuristic inspired considering a neutron traveling inside of a nuclear reactor, where two phenomena are noted: absorption, and scattering. The MPCA automatically finds the parameters associated to an ANN optimal architecture [1] by solving an optimization problem. The MPCA is an extension of the Particle Collision Algorithm (PCA). The MPCA is executed with a cooperative searching process to find the parameter values to optimize an objective function. The objective function which determines an MLP optimal architecture is given by

$$J(\mathbf{d}) = penality \left[\frac{\rho_1 E_{\text{train}}(\mathbf{d}) + \rho_2 E_{\text{gen}}(\mathbf{d})}{\rho_1 + \rho_2} \right] \tag{3}$$

$$penality = c_1 \, e^{\#\text{neurons}} + c_2 \, \{\#\text{epoch}\} + 1 \tag{4}$$

here $\mathbf{d}$ is the set of unknown parameters/functions to be computed, $\rho_1 = \rho_2 = 1$ are the adjustment parameters [10] for modifying the relevance between E_{train} (training error) and E_{gen} (generalization error), respectively. The *penality* term is metric of complexity of a neural network, and it is used to find the simplest architectures (lowest numbers of neurons, with fastest convergency for the computation of weights) for the MLP, with $c_1 = 5 \times 10^8$ and $c_2 = 5 \times 10^5$ [10].

2.2.2 Neural network implemented on FPGA

Two strategies were applied to enhance the processing to the computer vision with low power demand: storage the patterns identified by the MLP-NN on Look-up-Table (LUT), and secondly the hardware implementarion for the neural network by the FPGA (Field Programmable Gate Array). Two hardware arrangements were employed: Raspberry PI Model B-1(CPU: ARM1176JZF-S single core, 700 MHz com, 32-bits) with FPGA Spartan 6 LX9, and the Zybo Zynq 7000 – using *System-on-Chip* (SoC) technology with CPU ARM Cortex 9 dual core 650 MHz 32-bits and FPGA Artix-7.

3. Non-extensive Particle Filter and Data Fusion

The fundamental idea underlying the Sequential Monte Carlo is to represent the probability density function (PDF) by a set of samples with their associated weights. This set of samples is also referred to particles [**?**]. In the Particle Filter (PF), an estimation of *a posteriori* distribution is obtained by resampling with replacement from *a priori* ensemble. As the PF does not require assumptions of linearity or Gaussianityness, it is applicable to

general nonlinear problems. In particular, the PF can be applied to cases in which the relationship between a state and observed data is nonlinear.

Two properties are relevant issues for the PF: the Bayes theorem, and the Markov property. The Bayes theorem can express by the expression to calculate the conditional probability:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(A)} \tag{5}$$

where the probability $P(B)$ is understood as a normalization factor. The Markov process is characterized by the property:

$$p(u_n|u_{n-1}, \ldots, u_1, u_0) = p(u_n|u_{n-1}) \ . \tag{6}$$

For the current application, the distribution u_n represents the vector with the entries being the distributions associated to the UAV position estimations computed by visual odometry and computer vision system: the data fusion.

The algorithm for the PF implementation can be written as:

a) Compute the initial particle ensemble:

$$\left\{u_{0|n-1}^{(i)}\right\}_{i=0}^{M} \sim \rho_{u_0}(u_0)$$

the initial distribution ($p_{u_0}(u_0)$) is a free choice – our choice: $p_{u_0}(u_0) \sim N(0, 5)$ – the Gaussian distribution.

b) Compute:

$$r_n^{(i)} = p(z_n|u_{n|n-1}) = p_{\text{et}}(z_n - h(u_n, t_n))$$

where z_n denotes observations, and $p_{\text{et}}(z_n - h(u_n, t_n))$ is the likelihood function.

c) Normalize:

$$\hat{r}_n^{(i)} \frac{r_n^{(i)}}{\sum_{j=1}^{M} r_n^{(j)}} \ .$$

d) Resampling: extract particles with substitution, according to (a standard notation is used here – see [5, 12]):

$$Pr\left\{u_{n|n}^{(i)} = u_{n|n-1}^{(j)}\right\} = \hat{q}_n^{(j)} \ , \qquad i = 1, \ldots, M \ .$$

e) Time up-dating: compute the new particles:

$$u_{n+1|n}^{(i)} = f\left(u_{n+1|n}^{(i)}, t_n\right) + \mu_n \ , \quad \text{with: } \mu_n \in N(0,1)$$

where: $u_{n+1|n}^{(i)} \sim p\left(u_{n+1|n}^{(i)} | u_{n|n}^{(i)}\right)$, and $i = 1, \ldots, M$.

f) Set $t_{n+1} = t_n + \Delta t$, and go to the step (b).

The kernel coming from the application of the Bayes theorem and from the Markov property, suggesting the following choice:

$$\underbrace{p(u_n|z_n)}_{\text{posterior}} \propto \underbrace{p(z_n|u_n)}_{\text{likelihood}} \underbrace{p(u_n|z_{n-1})}_{\text{prior}} \ . \tag{7}$$

3.1 Non-extensive particle filter

In our approach, the likelihood operator is not a Gaussian function. In order to have a more flexible choice, we have employed the Tsallis' non-extensive form of entropy [13, 14].

$$S_q(p) = \frac{k}{q-1}\left[1 - \sum_{i=1}^{N} p_i^q\right] \tag{8}$$

where p_i is a probability, and q is a free parameter – called the non-extensivity parameter. In thermodynamics, the parameter k is known as the Boltzmanns constant. Tsallis entropy reduces to the the usual Boltzmann-Gibbs-Shanon formula in the limit $q \to 1$.

The equiprobability condition produces the maximum for the extensive and non-extensive forms to the entropy function, and this condition leads to the distributions [14]:

$$q > 1: \quad p_q(x) = \alpha_q^+ \left[1 - \frac{1-q}{3-q}\left(\frac{x}{\sigma}\right)^2\right]^{-1/(q-1)} \tag{9}$$

$$q = 1: \quad \frac{1}{\sigma}\left[\frac{1}{2\pi}\right] e^{-(x/\sigma)^2/2} \tag{10}$$

$$q < 1: \quad p_q(x) = \alpha_q^- \left[1 - \frac{1-q}{3-q}\left(\frac{x}{\sigma}\right)^2\right]^{1/(q-1)} \tag{11}$$

where

$$\sigma^2 \quad = \quad \frac{\int_{-\infty}^{+\infty} x^2 [p_q(x)]^q \, dx}{\int_{-\infty}^{+\infty} [p_q(x)]^q \, dx} \ ,$$

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

$$\alpha_q^+ = \frac{1}{\sigma}\left[\frac{q-1}{\pi(3-q)}\right]^{1/2}\frac{\Gamma(1/(q-1))}{\Gamma((3-q)/2(q-1))}\,,$$
$$\alpha_q^- = \frac{1}{\sigma}\left[\frac{1-q}{\pi(3-q)}\right]^{1/2}\frac{\Gamma((5-3q)/2(q-1))}{\Gamma((2-q)/(1-q))}\,.$$

The distributions above applies if $|x| < \sigma[(3-q)/(1-q)]^{1/2}$, and $p_q(x) = 0$ otherwise. For distributions with $q < 5/3$, the standard central limit theorem applies, implying that if p_q is written as a sum of M random independent variables, in the limit case $M \to \infty$, the probability density function for $p_q(x)$ in the distribution space is the normal (Gaussian) distribution. However, for $5/3 < q < 3$ the Levy-Gnedenko's central limit theorem applies, resulting for $M \to \infty$ the Lévy's distribution as the probability density function for the random variable p_q. The index in such Lévy distribution is $\gamma = (3-q)/(q-1)$ [14].

The purpose is to use the Tsallis' thermostatistics (9) or (11) for substituting the Gaussian function to represent the likelihood operator in step-(b) of the particle filter [5]. The idea is to explore the property of this thermostatistics to acess different attractors in the distribution space.

4. Drone Autonomous Navigation: FPGA and Non-extensive Particle Filter

The UAV images were obatined by helicopter RMAX (Yamaha Motor Company), used for testing in the Linköpin University (Sweden). The helicopter flew with average speed of 3 ms^{-1} and about 60 m over the surface (altitude). The UAV camera capture images from the Nadir direction with frequency of 25 Hz. The UAV resolution is 0.12 m/pixel with 288×360 pixels, and pixel corresponds an area 1540 m^2. The testing trajectory has about 1 km of extension, and 1443 images/points were obtained during the test.

Visual odometry and computer vision techniques has been applied to estimate drone or UAV position for autonomous navigation. Braga [3] did a study considering the latter techniques separately and as a combined methods – *data fusion*. A Particle Filter (PF) can be used to combine the cited methods – we are not assuming Gaussian hypothesis. In addition, different likelihood operators will go to produce better results to the PF estimation. Braga has used a non-extensive PF (NExt-PF) with 1000 particles to UAV positioning with visible band camera [3], doing a parametric investigation to determine the best value for the non-extensive parameter q, and he has obtained the value $q = 2.57$ [3].

The UAV positioning evaluation is analized by metrics adopted by Conte

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

e Doherty [4, 6] for comparison with the GPS positioning, where three metrics are employed: the Error Good Matching (EGM), the Standard Deviation Good Matching (SDGM), and Good Matching (GM). The EGM is the Euclidian distance between the estimated UAV position and that obtained by the GPS sensor on-board in the UAV. If the EGM is geather than 5 m, the estimative is not classified as an EGM, and it considered an EGM otherwise. The SDGM evaluates the method stability, and it is related with the standard deviation of the UAV positioning error considering the last 30 estimation results. If the error standard deviation is geather than 2 m, the estimation is not consideted as a SDGM. If the standard deviation is less than 2 m, the estimated trajectory is considered stable and the position is classified as a SDGM.

4.1 Computer vision using FPGA

For the visual odometry, the algorithm SURF was used to identify the points of interest, and the RANSAC (RANdom SAmple Consensus) filter [7] was applied to remove false corresponding points. For the computer vision system, a MLP neural network was trained to carry out the edges extraction for the objects in the images. The patterns used for image segmentation is shown in Figure 2, and Table 1 shows the MLP neural network topology determined by the MPCA optimizer.

Table 1: Neural network architecture and parameters computed from the MPCA meta-heuristics.

MLP-NN characteristics	Parameters/type
Neurons for the input layer	9
Neurons for the output layer	1
Number of hidden layers	1
Hidden layer neurons	18
Activation function	tanh
Learning rate	0.73
Momentum	0.85

Methods for edge extraction from images has different results related to the correctness and time processing execution. In addition, the MLP-NN was more effective than other two methods for edge identification – see Table 2. The time execution using different methods for edge identification is showb in Table 3 The best performance was obtained using the neural network described by LUT implemented on Zybo Zynq 7000 SoC (CPU+FPGA

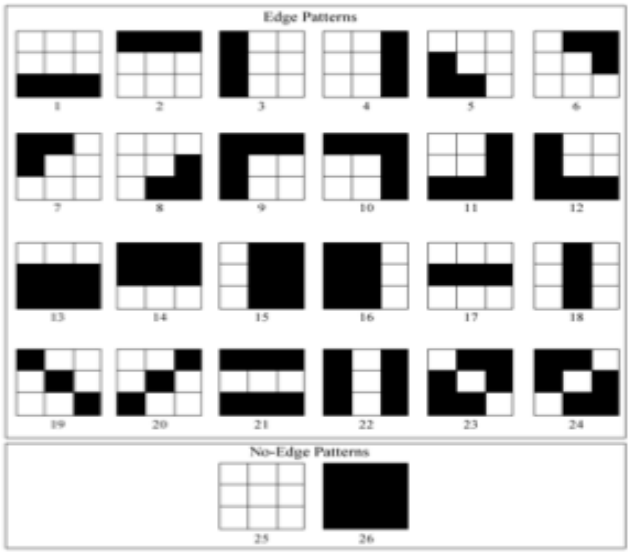

Figure 2: Patterns for MLP-NN learning phase.

in the same chip).

Table 2: EGM, SDGM, and GM metrics for edge extraction methods.

Method/evaluation	EGM	SDGM	GM
Canny	197	64	26
Sobel	296	63	42
MLP on LUT	414	216	103

4.2 Results with non-extensive particle filter

The estimated trajectory by data fusion is shown in Fig. 3 – blue line. The confidence interval can be found by Bayesian methods. The envelope for trajectory uncertainty quantification is also displayed in Fig. 3 – red lines.

Table 4 shows the evaluation for the UAV positioning according to the adopted metrics using computer vision system (CVS), visual odometry (VO), and data fusion (CVS+VO) by non-extensive particle filter with $q = 1.00$ (Gaussian function as likelihood operator), and $q = 2.51$. The worse result was obtained with visual odometry, and the best performance

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

Table 3: Methods for edge detection and execution time: CPU Raspberry (R), and CPU Zybo Zynq 7000 (Z).

Methods	Execution time (sec)
Canny – CPU (R/Z)	0.074 / 0.120
Sobel – CPU (R/Z)	0.083 / 0.140
Optimal MLP – CPU (R)	1.684
Optimal MLP – CPU (Z)	1.425
MLP on LUT – FPGA Spartan (R)	0.587
MLP on LUT – FPGA Altix (Z)	0.016

Table 4: EGM, SDGM, and GM metrics for the UAV positioning experiment.

Method/evaluation	EGM	SDGM	GM
CVS	926	1348	865
VO	667	1413	637
NExt-PF: $q = 1.00$	1303	1412	1273
NExt-PF: $q = 2.57$	1381	1413	1351

was the UAV trajectory determined by data fusion with non-extensive particle filter with $q = 2.57$.

5. Conclusions

The UAV positioning by image processing was effective using visual odometry (VO) and computer vision system (CVS). The VO has a disadvantage to present a cumulative error, and CVS needs a reference marks or images to estimate the UAV position. The data fusion approach can overcome the mentioned disadvantages.

The edge extraction by MLP-NN presented lower error than Canny and Sobel algorithms – see Table 2, but the neural network was more expensive in terms of the CPU-time. A LUT strategy was adopted to reduce the processing time. LUT with FPGA in SoC chip presented the faster processing time. Code optimization was not explored here. Therefore, there is a work to be done. A new approach for data fusion using the non-extensive particle was applied to the UAV positioning, with a better performance than standard particle filter. The method also allows to determine the confidence interval encapsulaning the uncertainties linked to the present estimation problem – Figure 3.

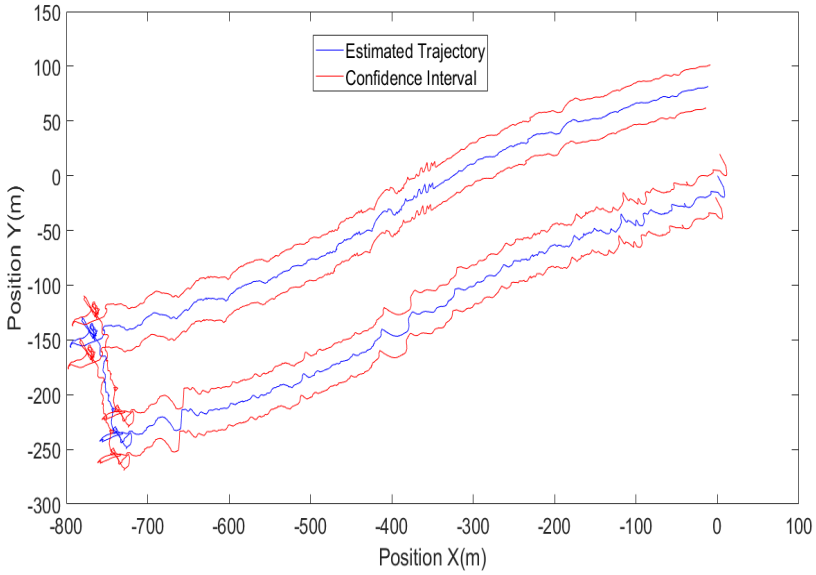

Figure 3: The confidence interval for the UAV trajectory estimated by non-extensive particle filter.

Acknowledgments: Authors want to thank to Fapesp, CNPq, and Capes, Brazilian agencies for research support.

References

[1] Anochi, J. A., Campos Velho, H. F., Furtado, H. C. M., Luz, E. F. P.: Self-configuring two types neural networks by MPCA. *2nd International Symposium on Uncertainty Quantification and Stochastic Modeling* (Uncertainties 2014), Rouen (France), 429-436, 2014.

[2] Bay, H., Ess, A., Tuytelaars, T, Gool, L.V.: Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3), 346-359, 2008.

[3] Braga, J. R. G.: *UAV Autonomous Navigation by LiDAR Images*, PhD thesis on Applied Computing, National Institute for Space Research (INPE), So Jos dos Campos (SP), Brazil, 2018.

[4] Braga J. R. G., Campos Velho, H. F., Conte, G., Doherty, P., Shiguemori E. H.: An image matching system for autonomous UAV navigation based on neural network. *14th International Conference on Control, Automation, Robotics and Vision* (ICARCV), Phuket (Thailand), 1-6, 2016.

[5] Campos Velho, H. F., Furtado, H. C. M.: Adaptive particle filter for stable distribution. *Integral Methods in Science and Engineering* (Eds: Christian Constanda and Paul J. Harris), Birkhäuser, 47-57, 2011.

[6] Conte, G., Doherty, P.: An integrated UAV navigation system based on aerial image matching. *IEEE Aerospace Conference, Big Sky* (MT, USA), 1-10, 2008.

[7] Fischler, M. A., Bolles R. C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24, 381-395, 1961.

[8] Gonzalez, R. C., Woods, R. E.: *Digital Image Processing* (4th Edition), Pearson, 2017.

[9] Haykin, S.: *Neural Networks: A Comprehensive Foundation*, Prentice Hall, 1998.

[10] Luz, E. F. P., Becceneri, J. C., Campos Velho, H. F.: A new multi-particle collision algorithm for optimization in a high performance environment. *Journal of Computational Interdisciplinary Sciences*, 1(1), 3-10, 2008.

[11] Nishar, A., Richrds, S., Breen, D., Robertson, J., Breen, B.: Thermal infrared imaging of geothermal environments and by an unmanned aerial vehicle (UAV): a case study of the wairakei-tauhara geothermal field, Taupo, New Zealand. *Renewable Energy*, 4(2), 136-145, 2016.

[12] Shon, T., Gustafsson, F., Nordlund, P.-J.: Marginalized particle filters for mixed linear/nonlinear state-space models. *IEEE Transactions on Signal Processing*, 53(7), 2279-2289, 2005.

[13] Tsallis, C.: Possible generalization of Boltzmann-Gibbs statistics". Journal of Statistical Physics, 52(1-2), 479-487, 1988.

[14] Tsallis, C.: Non additive entropy and non-extensive statistical mechanics – an overview after 20 years. *Brazilian Jounal of Physics*, 39(2A), 337-356, 2009.