Transactions Letters

Sum-Product Algorithm Utilizing Soft Distances on Additive Impulsive Noise Channels

M. Johnston, B. S. Sharif, *Senior Member, IEEE,* C. C. Tsimenidis, *Senior Member, IEEE*, and L. Chen, *Member, IEEE*

***Abstract*—In this letter, a Sum-Product algorithm (SPA) utilizing soft distances is shown to be more resilient to impulsive noise than conventional likelihood-based SPAs, when the noise distribution is unknown. An efficient version of the soft distance SPA is also developed but with half the storage requirements and running time.**

***Index Terms*—Block codes, decoding, impulse noise, iterative methods, Gaussian noise.**

I. INTRODUCTION

THE near-Shannon limit performance of low density parity check (LDPC) codes [1] is now well established and with the availability of efficient iterative belief propagation decoding algorithms [2] they are finding application in several emerging communication standards. Most of the existing literature on soft-decision decoding algorithms for LDPC codes assume that the noise present at the receiver has a Gaussian probability distribution, but in this letter we consider impulsive noise. In many applications Gaussian-distributed noise is a valid assumption but there are scenarios where a transmitted signal is also subject to man-made interference, e.g. underwater acoustic communication, urban environments and powerline communication. However, there is very little literature on the study of LDPC codes on impulsive noise channels. Maad *et al.* [3] investigated the performance of LDPC codes on impulsive noise channels with different de-mappers to counteract the impulses present in the background noise. Nakagawa *et al.* [4] modified the Sum-Product algorithm (SPA) by substituting the probability distribution function of the impulsive noise into the log-likelihood ratio calculation, i.e. assuming that the impulsive noise distribution is known at the receiver. Hu *et al.* [5] showed how to improve the performance of LDPC codes on impulsive noise channels by developing a low complexity method to estimate the variance of the noise. In this case, the measurement of the statistical properties of the noise is required before decoding.

Manuscript received July 4, 2011; revised August 22 and December 19, 2012. The editor coordinating the review of this paper and approving it for publication was K. Narayanan.

M. Johnston and C. Tsimenidis are with the School of Electrical and Electronic Engineering, Newcastle University, Newcastle-upon-Tyne, UK (e-mail: {martin.johnston, charalampos.tsimenidis}@ncl.ac.uk).

B. Sharif is with the Department of Electrical and Computer Engineering, Khalifa University, UAE (e-mail: bayan.sharif@kustar.ac.ae).

L. Chen is with the School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China (e-mail: chenli55@mail.sysu.edu.cn).

This work is partly supported by the National Natural Science Foundation of China (NSFC) with project ID: 61001094.

Digital Object Identifier 10.1109/TCOMM.2013.043013.110141

In 2008, Farrell introduced the idea of Soft Distance (SD) Decoding [6], which uses the squared Euclidean distances between each received symbol and all points in the constellation diagram. Soft distances are combined by taking the negative logarithm of the sum of anti-logarithms of the negative soft distances. In 2011, Farrell *et al.* [7] presented a soft distance sum-product algorithm (SD-SPA) that achieves the same level of performance as the conventional SPA on the additive white Gaussian noise (AWGN) channel. First impressions of soft distance decoding would suggest that there is no advantage over conventional likelihood-based soft-input-soft-output decoding since performance is the same on the AWGN channel. However, it was shown in [7] that the SD-SPA outperforms likelihood-based SPA decoding on a Rayleigh fading channel when the channel SNR is unknown. In this letter we investigate the performance and complexity of the SD-SPA, the logarithmic likelihood-based sum- product algorithm (log-SPA) and the Min-Sum algorithm (MSA) in the presence of impulsive noise with a Symmetric Alpha-Stable (SαS) distribution [8]. The MSA is a simplified version of the log-SPA but it is considered in this letter because, like the SD-SPA, it does not require knowledge of the channel. Furthermore, we also present a new version of the SD-SPA, called the Differential SD-SPA (DSD-SPA), which operates on the difference between soft distances rather than two separate soft distances. The letter is organized as follows: Section II gives a brief introduction to the SD-SPA and Section III focuses on the DSD-SPA. Section IV presents simulation results for the DSD-SPA, log-SPA and MSA on additive impulsive noise channels and conclusions are given in Section V.

II. ITERATIVE SUM-PRODUCT ALGORITHM WITH SOFT DISTANCES

An LDPC code is specified by its block length n, message length k, parity-check bit length $m = n - k$, row weight w_r and column weight w_c. The sparse parity-check matrix $\boldsymbol{H}$ comprises elements $h_{ij} \in GF(2)$, where $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$. For the remainder of this letter we use N_i to denote the set of indices where a 1 occurs in row i of $\boldsymbol{H}$, M_j to denote the set of indices where a 1 occurs in column j of $\boldsymbol{H}$ and the difference set $N_i \setminus j$ $(Mj \setminus i)$ to denote the

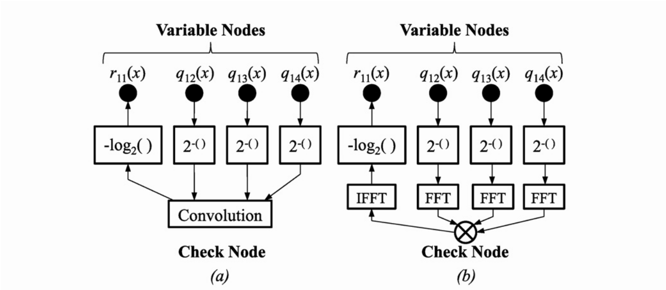

Fig. 1. (a) Example of the convolution operation and (b) the equivalent FFT operations at a hypothetical check node to determine $r_{11}(x)$.

elements in set N_i (M_j) excluding element j (i). Given a received vector $\boldsymbol{\rho} = [\rho_1, \rho_2, \dots \rho_n]$ the vector $\boldsymbol{d}$ contains the initial soft distances $d_j(x)$.

$$d_j(x) = (\rho_j - a)^2, j = 1, 2, \dots, n, \tag{1}$$

where $a = 2x - 1$ and $x \in GF(2)$. We define two $m \times n$ sparse matrices $\boldsymbol{Q_0}$ and $\boldsymbol{Q_1}$ with elements $q_{ij}(0)$ and $q_{ij}(1)$ respectively and initialize them as $q_{ij}(x) = d_j(x)$ when $h_{ij} = 1$.

A. Horizontal Step

The Horizontal step involves performing a convolutional operation at each check node in the Tanner graph. Two sparse $m \times n$ matrices $\boldsymbol{R_0}$ and $\boldsymbol{R_1}$ are defined comprising the soft distances $r_{ij}(0)$ and $r_{ij}(1)$ respectively. Two methods of applying the Horizontal step for the SD-SPA are illustrated in Fig. 1. In Fig. 1a, the soft distances $r_{11}(x)$ at one variable node connected to a check node of an arbitrary Tanner graph are determined by the convolution of the soft distances $q_{12}(x)$, $q_{13}(x)$ and $q_{14}(x)$ also connected to the same check node. However, the convolution operation at each check node generally becomes too complex for large row weights and so is replaced with more efficient FFTs and an inverse FFT (IFFT) [2] as shown in Fig. 1b. In $GF(2)$, the FFT and IFFT operations, F and F^{-1}, are both equivalent to multiplying by the Hadamard transform $\boldsymbol{W} = \frac{1}{\sqrt{2}}\left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$. The elements in $\boldsymbol{R_0}$ and $\boldsymbol{R_1}$ are determined by [7]

$$\begin{aligned} \begin{bmatrix} r_{ij}(0) \\ r_{ij}(1) \end{bmatrix} &= -\log_2 \left(F^{-1} \left\{ \prod_{j' \in N_i \setminus j} F \left\{ \begin{bmatrix} 2^{-q_{ij'}(0)}, \\ 2^{-q_{ij'}(1)} \end{bmatrix} \right\} \right\} \right) \\ &= -\log_2 \left(\frac{1}{2} \boldsymbol{W} \prod_{j' \in N_i \setminus j} \boldsymbol{W} \begin{bmatrix} 2^{-q_{ij'}(0)} \\ 2^{q_{ij'}(0)} \end{bmatrix} \right). \end{aligned} \tag{2}$$

Defining

$$2^{-\sigma q_{ij}} = 2^{-q_{ij}(0)} + 2^{-q_{ij}(1)}, \tag{3}$$

$$2^{-\delta q_{ij}} = 2^{-q_{ij}(0)} - 2^{-q_{ij}(1)}, \tag{4}$$

we finally obtain

$$\begin{aligned} r_{ij}(0) &= -\log_2 \left(\frac{1}{2} \left(\prod_{j' \in N_i \setminus j} 2^{-\sigma q_{ij}} + \prod_{j' \in N_i \setminus j} 2^{-\delta q_{ij}} \right) \right) \\ &= 1 - \log_2 \left(\prod_{j' \in N_i \setminus j} 2^{-\sigma q_{ij}} + \prod_{j' \in N_i \setminus j} 2^{-\delta q_{ij}} \right), \end{aligned} \tag{5}$$

$$\begin{aligned} r_{ij}(1) &= -\log_2 \left(\frac{1}{2} \left(\prod_{j' \in N_i \setminus j} 2^{-\sigma q_{ij}} - \prod_{j' \in N_i \setminus j} 2^{-\delta q_{ij}} \right) \right) \\ &= 1 - \log_2 \left(\prod_{j' \in N_i \setminus j} 2^{-\sigma q_{ij}} - \prod_{j' \in N_i \setminus j} 2^{-\delta q_{ij}} \right). \end{aligned} \tag{6}$$

B. Vertical Step

The Vertical step of the SD-SPA, where the matrices $\boldsymbol{Q_0}$ and $\boldsymbol{Q_1}$ are updated, is identical to the Vertical step in the log SPA and is a simple summation [7].

$$q_{ij}(x) = d_j(x) + \sum_{i' \in M_j \setminus i} r_{i'j}(x). \tag{7}$$

The values $q_{ij}(0)$ and $q_{ij}(1)$ can become too large with each decoding iteration, but this can be prevented by subtracting the smaller value of $q_{ij}(0)$ and $q_{ij}(1)$, i.e.

$$q_{ij}(x) = q_{ij}(x) - \min\{q_{ij}(0), q_{ij}(1)\}. \tag{8}$$

In a similar way to (7) the initial soft distances $\boldsymbol{d}$ are updated by [7]

$$d_j(x) = d_j(x) + \sum_{i \in M_j} r_{ij}(x). \tag{9}$$

If $d_j(0) < d_j(1)$ then the jth decoded bit is chosen to be $\hat{c}_j = 0$, otherwise $\hat{c}_j = 1$. The complete SD-SPA from [7] is summarized in Algorithm 1.

Algorithm 1: Soft Distance Sum-Product Decoding
Intialization
- Initialize $\boldsymbol{d}$ with the squared Euclidean distances $d_j(0)$ and $d_j(1)$ between the received symbol and both constellation points with (1)
- Obtain $\boldsymbol{Q_0}$ and $\boldsymbol{Q_1}$ by setting $q_{ij}(0) = d_j(0)$ and $q_{ij}(1) = d_j(1)$ when $h_{ij} = 1$, for $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$

Horizontal Step
- Determine all $r_{ij}(0)$ in $\boldsymbol{R_0}$ and $r_{ij}(1)$ in $\boldsymbol{R_1}$ using (5) and (6) respectively when $h_{ij} = 1$, for $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$

Vertical Step
- Update $\boldsymbol{Q_0}$ and $\boldsymbol{Q_1}$ using (7) and normalize the elements $q_{ij}(0)$ and $q_{ij}(1)$ with (8) when $h_{ij} = 1$, for $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$

Hard Decision
- Update soft distances $d_j(0)$ and $d_j(1)$ in $\boldsymbol{d}$ using (9), for $j = 1, 2, \dots, n$
- If $d_j(0) < d_j(1)$ then $\hat{c}_n = 0$ else $\hat{c}_n = 1$
- If $\hat{\boldsymbol{c}}\boldsymbol{H}^T = \boldsymbol{0}$ or a maximum number of decoding iterations have been reached then **stop**, otherwise continue to the **Horizontal Step**

III. Differential Soft Distance Sum-Product Algorithm

One disadvantage of the SD-SPA is the requirement to calculate and store two soft distances for each received symbol. However, by recognizing that soft distance decoding only depends on the difference between soft distances and not the individual soft distance values, we can halve storage requirements and consequently the running time. This is analogous to the log-SPA, which stores a single LLR instead of two probabilities. In this section, we present a version of the SD-SPA that uses and stores only the difference of the soft distances. We will refer to this algorithm as the differential SD-SPA (DSD-SPA). A vector $\boldsymbol{\Delta d}$ of length n, where each element Δd_j is the difference between the initial soft distances, is defined as

$$\Delta d_j = d_j(0) - d_j(1) = (\rho_j + 1)^2 - (\rho_j - 1)^2 = 4\rho_j. \quad (10)$$

As before, an $n \times m$ matrix ΔQ with elements Δq_{ij} is initialized by setting $\Delta q_{ij} = \Delta d_j$ when $h_{ij} = 1$, i.e. ΔQ is related to Algorithm 1 by $\boldsymbol{\Delta Q} = \boldsymbol{Q_0} - \boldsymbol{Q_1}$.

A. New Horizontal Step

In reducing the running time and storage requirements of the Horizontal Step, our main goal was to express the difference of the soft distances $\Delta r_{ij} = r_{ij}(0) - r_{ij}(1)$ in terms of Δq_{ij}. Since it is the difference in soft distances that we are interested in, we can assume, without loss of generality, that for all received symbols either $q_{ij}(0)$ or $q_{ij}(1)$ is zero. If $q_{ij}(1) = 0$ then $\Delta q_{ij} = q_{ij}(0)$ and substituting into (4) and (5) we obtain

$$2^{-\sigma q_{ij}} = 2^{-q_{ij}(0)} + 1 = 2^{-|\Delta q_{ij}|} + 1,$$
$$2^{-\delta q_{ij}} = 2^{-q_{ij}(0)} - 1 = 2^{-|\Delta q_{ij}|} - 1.$$

Similarly, if $q_{ij}(0) = 0$ then $\Delta q_{ij} = -q_{ij}(1)$ and

$$2^{-\sigma q_{ij}} = 2^{-q_{ij}(1)} + 1 = 2^{-|\Delta q_{ij}|} + 1,$$
$$2^{-\delta q_{ij}} = 1 - 2^{-q_{ij}(1)} = 1 - 2^{-|\Delta q_{ij}|}.$$

Combining these expressions for all positive and negative values of Δq_{ij} we obtain

$$2^{-\sigma q_{ij}} = 2^{-|\Delta q_{ij}|} + 1. \quad (11)$$

$$2^{-\delta q_{ij}} = \text{sign}(\Delta q_{ij}) \left(2^{-|\Delta q_{ij}|} - 1\right). \quad (12)$$

We define an $m \times n$ matrix $\boldsymbol{\Delta R}$ with elements $\Delta r_{ij} = r_{ij}(0) - r_{ij}(1)$ and subtract (6) from (5) to obtain

$$\Delta r_{ij} = -\log_2 \left(\frac{\prod_{j' \in N_i \setminus j} 2^{-\sigma q_{ij'}} + \prod_{j' \in N_i \setminus j} 2^{-\delta q_{ij'}}}{\prod_{j' \in N_i \setminus j} 2^{-\sigma q_{ij'}} - \prod_{j' \in N_i \setminus j} 2^{-\delta q_{ij'}}} \right). \quad (13)$$

Finally, substituting (11) and (12) into (13) gives us an expression for Δr_{ij} in terms of Δq_{ij} as shown in (14).

B. New Vertical Step and Hard Decision

The Vertical Step will update each element Δq_{ij} in $\boldsymbol{\Delta Q}$ using the elements Δr_{ij}. Since $\Delta q_{ij} = q_{ij}(0) - q_{ij}(1)$ then

$$\Delta q_{ij} = (d_j(0) - d_j(1)) + \sum_{i' \in M_j \setminus i} r_{i'j}(0) - r_{i'j}(1). \quad (15)$$

This simplifies to

$$\Delta q_{ij} = 4\rho_j + \sum_{i' \in M_j \setminus i} \Delta r_{i'j}, \quad (16)$$

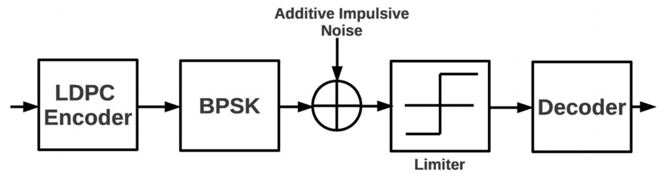

Fig. 2. System model.

thus removing the normalization step in (8). Finally, the initial soft distances Δd_j are updated by

$$\Delta d_j = 4\rho_j + \sum_{i \in M_j} \Delta r_{ij}. \quad (17)$$

If $\Delta d_j < 0$ then the jth decoded bit is chosen to be $\hat{c}_n = 0$, otherwise $\hat{c}_n = 1$.

Algorithm 2: Differential Soft Distance Sum-Product Decoding

Intialization

- Initialize $\boldsymbol{\Delta d}$ by calculating Δd_j using (10), for $j = 1, 2, \ldots, n$
- Obtain $\boldsymbol{\Delta Q}$ by setting $\Delta q_{ij} = \Delta d_j$ when $h_{ij} = 1$, for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$

Horizontal Step

- Determine all Δr_{ij} using (14) when $h_{ij} = 1$, for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$

Vertical Step

- Update Δq_{ij} using (16) when $h_{ij} = 1$, for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$

Hard Decision

- Update Δd_j using (17), for $j = 1, 2, \ldots, n$
- If $\Delta d_j < 0$ then $\hat{c}_n = 0$ else $\hat{c}_n = 1$
- If $\hat{\boldsymbol{c}}\boldsymbol{H}^T = \boldsymbol{0}$ or a maximum number of decoding iterations have been reached then **stop**, otherwise continue to the **Horizontal Step**

IV. Results

The performances of the DSD-SPA, log-SPA and MSA are evaluated using the (2304, 1152) irregular quasi-cyclic LDPC code as specified in the IEEE802.16e-2005 standard. The system model is illustrated in Fig.2. The encoded message is mapped to Binary Phase Shift Keying (BPSK) symbols and impulsive noise, with an SαS distribution and characteristic exponent values of $\alpha = 0.5$, 1 and 2, is added. The received signal is clipped to reduce the magnitude of the impulses and then iteratively decoded, with a maximum number of iterations set to 20. For fairness the log-SP decoder has knowledge of the noise variance, which is measured from the received values. Fig. 3 shows the bit-error rate (BER) performance of the LDPC code with the DSD-SPA, the log-SPA and MSA on three different additive impulsive noise channels as a function of the geometric signal-to-noise ratio [9]. When $\alpha = 2$ the noise has a Gaussian distribution and the performance of the DSD-SPA and log-SPA are the same, while the performance of the MSA is slightly degraded as expected. When $\alpha = 1$ the channel is more impulsive and we observe

$$\Delta r_{ij} = -\log_2 \left(\frac{\prod_{j' \in N_i \backslash j} 2^{-|\Delta q_{ij'}|} + 1 + \prod_{j' \in N_i \backslash j} \text{sign}(\Delta q_{ij'}) \left(2^{-|\Delta q_{ij'}|} - 1\right)}{\prod_{j' \in N_i \backslash j} 2^{-|\Delta q_{ij'}|} + 1 - \prod_{j' \in N_i \backslash j} \text{sign}(\Delta q_{ij'}) \left(2^{-|\Delta q_{ij'}|} - 1\right)} \right). \tag{14}$$

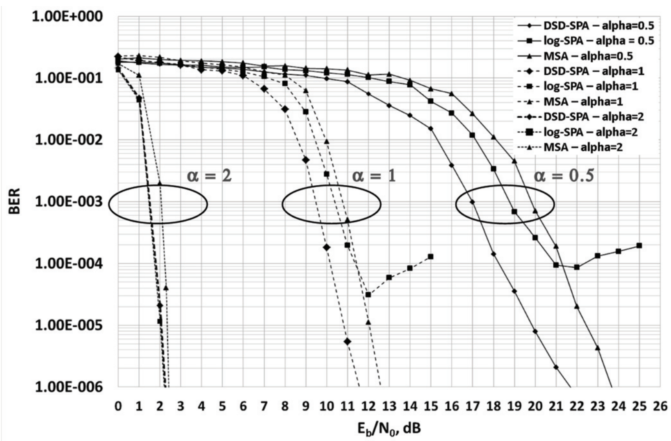

Fig. 3. BER performance of the (2304, 1152) irregular QC-LDPC code on additive impulsive noise channels with $\alpha = 0.5, 1$ and 2.

that the performance of the log-SPA seriously degrades from E_b/N_0 = 12 dB due to the incorrect assumption that the noise is Gaussian. However, the DSD-SPA and MSA perform better and do not exhibit error floors up to a BER of 10^{-6}. The MSA outperforms the log-SPA in this situation as it also does not require knowledge of the noise distribution. Nevertheless, the DSD-SPA still achieves a coding gain of approximately 1dB over the MSA at a BER = 10^{-6}. Finally, when $\alpha = 0.5$ the channel is extremely impulsive and we observe the log-SPA causes an error floor from E_b/N_0 = 20 dB at a BER $\simeq 10^{-4}$. The DSD-SPA and MSA again perform significantly better, but the DSD-SPA achieves a coding gain of around 2dB over the MSA at a BER = 10^{-6}.

V. Conclusion

In this letter, a new decoding algorithm based on SD-SPA, called the DSD-SPA has been presented. By operating on single values corresponding to differences in soft distances rather than individual soft distance values the number of matrices needed is halved and running time is reduced. Additionally, the DSD-SPA suffers no degradation in performance and achieves the same complexity as the log-SPA. Simulation results of a long LDPC code have been presented on additive impulsive noise channels and show the DSD-SPA is significantly more resilient to impulses compared with the log-SPA. The MSA also does not rely on knowledge of the channel and it too performed significantly better than the log-SPA. However, the DSD-SPA still outperformed the MSA by approximately 1dB and 2dB on an impulsive noise channel with $\alpha = 1$ and 0.5 respectively. This improved performance arises as a result of the SD-SPA and DSD-SPA creating a locally determined variance from the initial soft distances of the received symbols, unlike likelihood-based decoders. The DSD-SPA is analogous to the log-SPA, processing differential soft distances instead of LLRs. It has similar Horizontal and Vertical steps and therefore, has essentially the same complexity. By using soft distances as a decoding metric we believe the DSD-SPA is an attractive and practical alternative decoder that could be deployed in any environment.

References

[1] R. G. Gallager, *Low Density Parity Check Codes*. MIT Press, 1963.
[2] T. K. Moon, *Error Correction Coding: Mathematical Methods and Algorithms*. Wiley-Interscience, 2005.
[3] H. B. Maad, A. Goupil, L. Clavier, and G. Gelle, "Asymptotic performance of ldpc codes in impulsive non-Gaussian channel," in *Proc. 2010 Int. Workshop on Signal Processing Advances in Wireless Communications*, pp. 1–5.
[4] H. Nakagawa, D. Umehara, S. Denno, and Y. Morihiro, "A decoding for low density parity check codes over impulsive noise channels," in *Proc. 2005 International Symposium on Power Line Communications and its Applications*, pp. 85–89.
[5] C. Hu, N. Weng, W.-Y. Chan, and P. Jain, "Improving a power line communications standard with LDPC codes," *EURASIP J. Advances Signal Process.*, vol. 2007, no. 60839.
[6] P. G. Farrell, "Decoding error control codes with soft distance as the metric," in *2008 Workshop on Mathematical Techniques in Coding Theory*.
[7] P. G. Farrell, L. Arnone, and J. C. Moreira, "Euclidean distance soft-input soft-output decoding algorithm for low-density parity-check codes," *IET Commun.*, vol. 5, no. 16, p. 2364, 2011.
[8] J. P. Nolan, "Stable distributions: models for heavy tailed data." Available: http://academic2.american.edu/ jpnolan/stable.html.
[9] J. G. Gonzalez, D. W. Griffith, and G. R. Arce, "Zero-order statistics: a signal processing framework for very impulsive environments," in *Proc. 1997 Signal Processing Workshop on Higher Order Statistics*.