

# New Method for Filtered ICA Signals Applied To Volatile Time Series.

J.M. Górriz, Carlos G. Puntonet, Moisés Salmerón, and Julio Ortega

Dpto. Ingeniería de Sistemas y Automática,  
Tec. Electrónica y Electrónica. Universidad de Cádiz, and  
Dpto. Arquitectura y Tecnología de Computadores,  
Universidad de Granada  
E-mail: [juanmanuel.gorritz@uca.es](mailto:juanmanuel.gorritz@uca.es)

**Abstract.** In this paper we propose a new method for volatile time series forecasting using techniques like Independent Component Analysis (ICA) or Savitzky-Golay filtering as preprocessing tools. The preprocessed data will be introduced in a based radial basis functions (RBF) Artificial Neural Network (ANN) and the prediction result will be compared with the one we get without these preprocessing tools or the classical Principal Component Analysis (PCA) tool.

## 1 Introduction

Different techniques have been developed in order to forecast time series using data from the stock. There also exist numerous forecasting applications like those ones analyzed in [17]: signal statistical preprocessing and communications, industrial control processing, Econometrics, Meteorology, Physics, Biology, Medicine, Oceanography, Seismology, Astronomy y Psychology.

A possible solution to this problem was described by Box and Jenkins [7]. They developed a time-series forecasting analysis technique based in linear systems. Basically the procedure consisted in suppressing the non-seasonality of the series, parameters analysis, which measure time-series data correlation, and model selection which best fitted the data collected (some specific order ARIMA model). But in real systems non-linear and stochastic phenomena crop up, thus the series dynamics cannot be described exactly using those classical models. ANNs have improved results in forecasting, detecting the non-linear nature of the data. ANNs based in RBFs allow a better forecasting adjustment; they implement local approximations to non-linear functions, minimizing the mean square error to achieve the adjustment of neural parameters. Platt's algorithm [16], RAN (Resource Allocating Network), consisted in the control of the neural network's size, reducing the computational cost associated to the calculus of the optimum weights in perceptrons networks. Matrix decomposition techniques have been used as an improvement of Platt model [24] with the aim of taking the most relevant data in the input space, for the sake of avoiding the processing of non-relevant information (NAPA-PRED "Neural model with Automatic Parameter Adjustment for PREDiction"). NAPA-PRED also includes neural pruning [23].

The next step was to include the exogenous information to these models. There are some choices in order to do that; we can use the forecasting model used in [9] which gives good results but with computational time and complexity cost; Principal Component Analysis (PCA) is a well-established tool in Finance. It was already proved [24] that prediction results can be improved using the PCA technique. This method linear transform the observed signal into principal components which are uncorrelated (features), giving projections of the data in the direction of the maximum variance [15]. PCA algorithms use only second order statistical information; Finally, in [3] we can discover interesting structure in finance using the new signal-processing tool Independent Component Analysis (ICA). ICA finds statistically independent components using higher order statistical information for blind source separation ([4], [13]). This new technique may use Entropy (Bell and Sejnowski 1995, [6]), Contrast functions based on Information Theory (Comon 1994, [8]), Mutual Information (Amari, Cichocki y Yang 1996, [2]) or geometric considerations in data distribution spaces (Carlos G. Puntonet 1994 [19],[26], [1], [20], [21]), etc. Forecasting and analyzing financial time series using ICA can contributes to a better understanding and prediction of financial markets ([22],[5]).

## 2 Basic ICA

ICA has been used as a solution of the blind source separation problem of a linear mixture [12] denoting the process of taking a set of measured signal in a vector,  $\mathbf{x}$ , and extracting from them a new set of statistically independent components (ICs) in a vector  $\tilde{\mathbf{s}}$ . In the basic ICA each component of the vector  $\mathbf{x}$  is a linear instantaneous mixture of independent source signals in a vector  $\mathbf{s}$  with some unknown deterministic mixing coefficients:

$$x_i = \sum_{j=1}^N a_{ij}s_j \tag{1}$$

Due to the nature of the mixing model we are able to estimate the original sources  $\tilde{s}_i$  and the de-mixing weights  $b_{ij}$  applying i.e. ICA algorithms based on higher order statistics like cumulants.

$$\tilde{s}_i = \sum_{j=1}^N b_{ij}x_j \tag{2}$$

Using vector-matrix notation and defining a time series vector  $\mathbf{x} = (x_1, \dots, x_n)^T$ ,  $\mathbf{s}$ ,  $\tilde{\mathbf{s}}$  and the matrix  $\mathbf{A} = \{a_{ij}\}$  and  $\mathbf{B} = \{b_{ij}\}$  we can write the overall process as:

$$\tilde{\mathbf{s}} = \mathbf{B}\mathbf{x} = \mathbf{B}\mathbf{A}\mathbf{s} = \mathbf{G}\mathbf{s} \tag{3}$$

where we define  $\mathbf{G}$  as the overall transfer matrix. The estimated original sources will be, under some conditions included in Darmois-Skitovich theorem, a permuted and scaled version of the original ones. Thus, in general, it is only possible

to find  $\mathbf{G}$  such that  $\mathbf{G} = \mathbf{PD}$  where  $\mathbf{P}$  is a permutation matrix and  $\mathbf{D}$  is a diagonal scaling matrix.

In Financial time series this model (equation 1) can be applied to the stock series where there are some underlying factors like seasonal variations or economic events that affect the stock time series simultaneously and can be assumed to be quite independent [14].

### 3 Preprocessing Volatile Signals

The main goal, in the preprocessing step, is to find non-volatile time series including exogenous information i.e. financial time series, easier to predict using ANNs based on RBFs. This is due to smoothed nature of the kernel functions used in regression over multidimensional domains [11]. We propose the following Preprocessing Steps

- After Whitening the set of time series  $\{x_i\}_{i=1}^n$  (subtract the mean of each time series and removing the second order statistic effect or covariance matrix diagonalization process)
- we apply some ICA algorithm to estimate the original sources  $s_i$  and the mixing matrix  $\mathbf{A}$  in equation 1. Each IC has information of the stock set weighted by the components of the mixing matrix. In particular, we use an equivariant robust ICA algorithm based in cumulants. The de-mixing matrix is calculated according the following iteration:

$$\mathbf{B}^{(n+1)} = \mathbf{B}^{(n)} + \mu^{(n)}(\mathbf{C}_{\mathbf{s},\mathbf{s}}^{1,\beta} \mathbf{S}_{\mathbf{s}}^{\beta} - \mathbf{I})\mathbf{B}^{(n)} \tag{4}$$

where  $\mathbf{I}$  is the identity matrix,  $\mathbf{C}_{\mathbf{s},\mathbf{s}}^{1,\beta}$  is the  $\beta + 1$  order cumulant of the sources (we chose  $\beta = 3$  in simulations) and  $\mathbf{S}_{\mathbf{s}}^{\beta} = \text{diag}(\text{sign}(\text{diag}(\mathbf{C}_{\mathbf{s},\mathbf{s}}^{1,\beta})))$ . Once convergence, which is related with cross-cumulants <sup>1</sup> absolute value, is reached , we estimate the mixing matrix inverting  $\mathbf{B}$ .

- Filtering.
  1. We neglect non-relevant components in the mixing matrix  $\mathbf{A}$  according to their absolute value. We consider the rows  $\mathbf{A}_i$  in matrix  $\mathbf{A}$  as vectors and calculate the mean Frobenius Norm <sup>2</sup> of each one. Only the components bigger than mean Frobenius Norm will be considered. This is the principal preprocessing step using PCA tool, in this case is not enough.

$$\tilde{\mathbf{A}} = \mathbf{Z} \cdot \mathbf{A} \tag{5}$$

where  $\{\mathbf{Z}\}_{ij} = [\{\mathbf{A}\}_{ij} > \frac{\|\mathbf{A}_i\|_{Fr}}{n}]$

2. We apply a low band pass filter to the ICs. We choose the well-adapted for data smoothing Savitsky-Golay smoothing filter [25] for two reasons:
  - a)ours is a real-time application for which we must process a continuous

<sup>1</sup> cumulants between different sources

<sup>2</sup> Frobenius Norm:  $\|\mathbf{x}\|_{Fr} \equiv \sqrt{\sum_{i=1}^n x_i^2}$

data stream and wish to output filtered values at the same rate we receive raw data and *b*) the quantity of data to be processed is so large that we just can afford only a very small number of floating operations on each data point thus computational cost in frequency domain for high dimensional data is avoided even the modest-sized FFT. This filter is also call Least-Squares [10], and derives from a particular formulation of the data smoothing problem in the time domain and their goal is to find filter coefficients  $c_n$  in the expression:

$$\bar{s}_i = \sum_{n=-n_L}^{n_R} c_n s_{i+n} \tag{6}$$

where  $\{s_{i+n}\}$  represent the values for the ICs in a window of length  $n_L + n_R + 1$  centered on  $i$  and  $\bar{s}_i$  is the filter output (the smoothed ICs), preserving higher moments [18]. For each point  $s_i$  we least-squares fit a  $m$  order polynomial for all  $n_L + n_R + 1$  points in the moving window and then set  $\bar{s}_i$  to the value of that polynomial at position  $i$ . As shown in [18] there are a set of coefficients for which equation 6 accomplishes the process of polynomial least-squares fitting inside a moving window:

$$c_n = \{(\mathbf{M}^T \cdot \mathbf{M})^{-1}(\mathbf{M}^T \cdot \mathbf{e}_n)\}_0 = \sum_{j=0}^m \{(\mathbf{M}^T \cdot \mathbf{M})^{-1}\}_{0j} \cdot n^j \tag{7}$$

where  $\{\mathbf{M}\}_{ij} = i^j$ ,  $i = -n_L, \dots, n_R$ ,  $j = 0, \dots, m$ , and  $\mathbf{e}_n$  is the unit vector with  $-n_L < n < n_R$ .

- Reconstructing the original signals using the smoothed ICs and filtered  $\tilde{\mathbf{A}}$  matrix we get a less high frequency variance version including exogenous influence of the old ones. We can write using equations 5 and 4.

$$\mathbf{x} = \tilde{\mathbf{A}} \cdot \bar{\mathbf{s}} \tag{8}$$

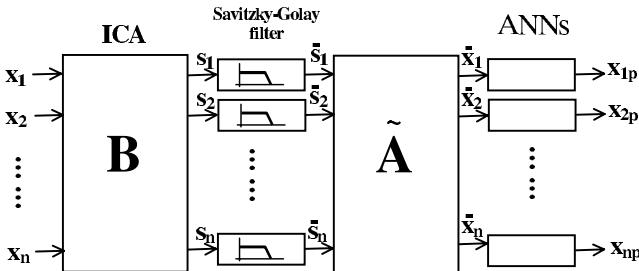


Fig. 1. Schematic representation of prediction and filtering process.

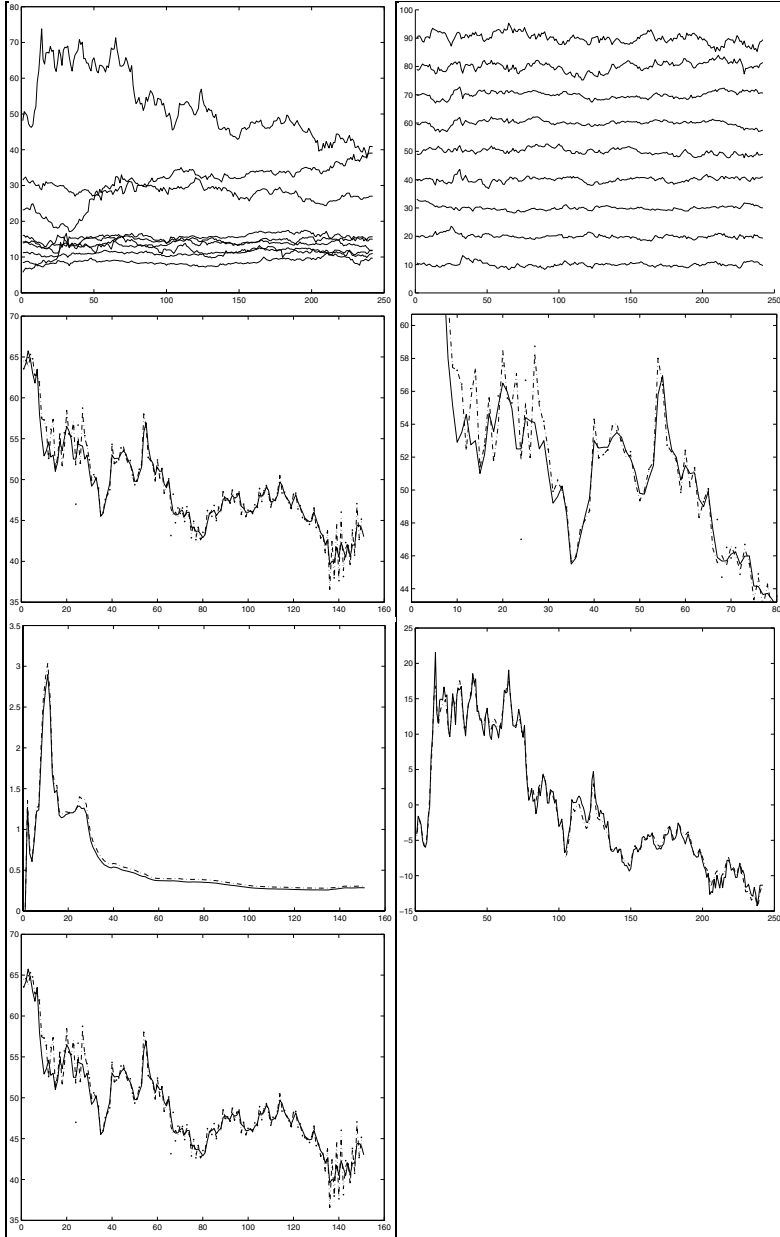
## 4 Time Series Forecasting Model

We use an ANN using RBFs to forecast a series  $x_i$  from the Stock Exchange building a forecasting function  $\mathbf{P}$  with the help of NAPA-PRED+LEC algorithm, for one of the set of signals  $\{x_1, \dots, x_n\}$ . The individual forecasting function can be expressed in term of RBFs [27] as:  $\mathbf{F}(\mathbf{x}) = \sum_{i=1}^N f_i(\mathbf{x}) = \sum_{i=1}^N h_i \exp\left\{-\frac{\|\mathbf{x}-\mathbf{c}_i\|^2}{r_i^2}\right\}$ , where  $\mathbf{x}$  is a p-dimensional vector input at time  $\mathbf{t}$ ,  $\mathbf{N}$  is the number of neurons (RBFs),  $f_i$  is the output for each neuron i-th,  $\mathbf{c}_i$  is the centers of i-th neuron which controls the situation of local space of this cell and  $r_i$  is the radius of the i-th neuron. The global output is a linear combination of the individual output for each neuron with the weight of  $\mathbf{h}_i$ . Thus we are using a method for moving beyond the linearity where the core idea is to augment/replace the vector input  $\mathbf{x}$  with additional variables, which are transformations of  $\mathbf{x}$ , and then use linear models in this new space of derived input features. RBFs are one of the most popular kernel methods for regression over the domain and consist on fitting a different but simple model at each query point  $\mathbf{c}_i$  using those observations close to this target point in order to get a smoothed function. This localization is achieved via a weighting function or kernel  $f_i$ . The preprocessing step suggested in section 3 is necessary due to the dynamic of the series [22] and it will be shown that results improve sensitively. Thus we will use as input series the ones we got in equation 8.

## 5 Simulations and Conclusions.

We work with indexes of different Spanish banks and companies during the same period to investigate the effectiveness of ICA techniques for financial time series (1 in table 2). We have specifically focussed on the IBEX35 from Spanish stock, which we consider the most representative sample of the Spanish stock movements, using closing prices in 2000. We considered the closing prices of Bankinter for prediction and 8 more stock of different Spanish companies (ACS, Aguas de Barcelona, Banco Popular, Banco Santander, BBVA, Dragados, Carrefour and Amadeus). Each time series includes 200 points corresponding to selling days (quoting days). We performed ICA on the Stock returns using the ICA algorithm presented in section 3 assuming that the number of stocks equals the number of sources supplied to the mixing model. This algorithm whiten the raw data as the first step. The ICs are shown in 2 table 2. These ICs represents independent and different underlying factors like seasonal variations or economic events that affect the stock time series simultaneously. Via the rows of  $\mathbf{A}$  we can reconstruct the original signals with the help of these ICs i.e. by Frobenius filtering the original mixing matrix is transformed, and we neglect the influence of the corresponding ICs on the original stock. Thus only a few ICs contribute to most of the movements in the stock returns and each IC contributes to a level change depending its amplitude transient [3]. We do a polynomial fit in the ICs using the library supported by *MatLab* and the reconstruct the selected stock (6 and 7 in table 2) to supply the ANN.

**Fig. 2.** Simulations. 1 Set of Stock Series. 2 ICs for the stock series. 3 Real Series(line) Predicted Series with ICA+SG(-) Predicted Series without preprocessing (.). 4 Zoom 3. 5 NRMSE evolution for ANN with ICA+SG (line) and without (-). 6 Real Series from ICA reconstruction (scaled old version)(line) and Preprocessed Real Series (-). 7 Zoom 6.



In 3,4 and 5 in table 2 we show the results we got using our ANN with the above mentioned algorithm. We can say that prediction is better with the preprocessing step avoiding the disturbing peaks or convergence problems in prediction. As is shown in 5 the NRMSE is always lower using the techniques we discussed in section 4.

## 5.1 Conclusions.

In this paper we showed that prediction results can be improved with the help of techniques like ICA. ICA decompose a set of 9 returns from the stock into independent components which fall in two categories: *a)* large components responsible of the major changes in level prices and *b)* small fluctuations. Smoothing this components and neglecting the non-relevant ones we can reconstruct a new version of the Stock easier to predict. Moreover we describe a new filtering method to volatile time series that are supplied to ANNs in real-time applications.

## 6 Acknowledgments

This research was partially supported by Spanish CICYT Projects TIC2000-1348 and TIC2001-2845.

## References

1. C.G. Puntonet A. Mansour, N. Ohnishi, *Blind multiuser separation of instantaneous mixture algorithm based on geometrical concepts*, Signal Processing **82** (2002), 1155–1175.
2. S. Amari, A. Cichocki, and H. Yang, *A new learning algorithm for blind source separation*, Advances in Neural Information Processing Systems. MIT Press **8** (1996), 757–763.
3. A. D. Back and A. S. Weigend, *Discovering structure in finance using independent component analysis*, Computational Finance (1997).
4. Andrew D. Back and Thomas P. Trappenberg, *Selecting inputs for modelling using normalized higher order statistics and independent component analysis*, IEEE Transactions on Neural Networks **12** (2001).
5. Andrew D. Back and A.S. Weigend, *Discovering structure in finance using independent component analysis*, 5th Computational Finance 1997 (1997).
6. A.J. Bell and T.J. Sejnowski, *An information-maximization approach to blind separation and blind deconvolution*, Neural Computation **7** (1995), 1129–1159.
7. G.E.P. Box, G.M. Jenkins, and G.C. Reinsel, *Time series analysis. forecasting and control*, Prentice Hall, 1994.
8. P. Comon, *Independent component analysis: A new concept?*, Signal Processing **36** (1994), 287–314.
9. J.M. Górriz, J.J.G. dela Rosa, Carlos G. Puntonet, and M. Salmerón Campos, *New model for time-series forecasting using rbf's and exogenous data*, In Press (2003).
10. R.W. Hamming, *Digital filters*, 2<sup>a</sup> ed., Prentice Hall, 1983.
11. T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning*, Springer, 2000.

12. A. Hyvärinen, J. Karhunen, and E. Oja, *Independent component analysis*, John Wiley and Sons, 2001.
13. A. Hyvärinen and E. Oja, *Independent component analysis: Algorithms and applications*, Neural Networks **13** (2000), 411–430.
14. K. Kiviluoto and E. Oja, *Independent component analysis for paralell financial time series*, Proc. in ICONIP98 **1** (1998), 895–898.
15. T. Masters, *Neural, novel and hybrid algorithms for time series analysis prediction*, John Miley & Sons, 1995.
16. J. Platt, *A resource-allocating network for function interpolation*, Neural Computation **3** (1991), 213–225.
17. D.S.G. Pollock, *A handbook of time series analysis, signal processing and dynamics*, Academic Press, 1999.
18. W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical recipes in c++*, 2<sup>a</sup> ed., Cambridge University Press, 2002.
19. C.G. Puntonet, *Nuevos Algoritmos de Separación de Fuentes en Medios Lineales*, Ph.D. thesis, University of Granada, Departamento de Electrónica y Tecnología de Computadores, 1994.
20. C.G. Puntonet and Ali Mansour, *Blind separation of sources using density estimation and simulated annealing*, IEICE Transactions on Fundamental of Electronics Communications and Computer Sciences **E84-A** (2001).
21. M. Rodríguez-Álvarez, C.G. Puntonet, and I. Rojas, *Separation of sources based on the partitioning of the space of observations*, Lecture Notes in Computer Science **2085** (2001), 762–769.
22. J.M. Górriz Sáez, *Predicción y Técnicas de Separación de Senales*, Ph.D. thesis, University of Cádiz, Departamento de Ing. de Sistemas y Aut. Tec. Electrónica y Electrónica, 2003.
23. Moisés Salmerón, Julio Ortega, Carlos G. Puntonet, and Alberto Prieto, *Improved ran sequential prediction using orthogonal techniques*, Neurocomputing **41** (2001), 153–172.
24. M. Salmerón-Campos, *Predicción de Series Temporales con Redes Neuronales de Funciones Radiales y Técnicas de Descomposición Matricial*, Ph.D. thesis, University of Granada, Departamento de Arquitectura y Tecnología de Computadores, 2001.
25. A. Savitzky and M.J.E. Golay, *Analytical Chemistry* **36** (1964), 1627–1639.
26. F. J Theis, A. Jung, E.W. Lang, and C.G. Puntonet, *Multiple recovery subspace projection algorithm employed in geometric ica*, In press on Neural Computation (2001).
27. J. Moody and C. J. Darken, *Fast learning in networks of locally-tuned processing units*, Neural Computation **1** (1989), 284–294.