

# Independent Component Analysis for Radio Network Prediction Enhancement

Zakaria Nour, Berna Sayrac and Benoît Fourestié

France Telecom, Research and Development Division

38 rue Général Leclerc Issy-les-Moulineaux FRANCE

E-mail : (zakaria.nour,berna.sayrac,benoit.fourestie)@francetelecom.com

Walid Tabbara and Françoise Brouaye

Departement de Recherche en Electromagnétisme

3, rue Joliot-Curie Gif sur Yvette FRANCE

E-mail : (tabbara,brouaye)@lss.supelec.fr

**Keywords**—Artificial Neural Networks, Independent Component Analysis, Radio Network Prediction, Hybrid Simulation.

**Abstract**— We propose a method to enhance the quality and precision of prediction results using measurements in the context of radio network modelling. The proposed method involves the use of an Independent Component Analysis (ICA) block and a MultiLayer Perceptron (MLP) Artificial Neural Network (ANN). The role of the ICA block is to make the variables at the input of the ANN statistically independent so that it can perform its learning and prediction on individual one-dimensional distributions.

The application of the proposed method to a third generation cellular radio network prediction tool has shown that without ICA, ANN training has a poor performance. We have also shown that, in the proposed scheme, ICA performs better than Principle Component Analysis (PCA). This enhancement method can advantageously be used to calibrate prediction results according to measurements.

## I. INTRODUCTION

To simulate a complex system one can resort to two types of simulation principles. The first is based on the use of the *a priori* knowledge of the system at hand, i.e. the real physical phenomena are represented by mathematical models [Centeno and Reyes, 1998] (see figure 1). The second is based on the use of the *a posteriori* information, i.e. measured data characterizing the real system, to train an automatic learning system such as a neural network [Andrea et al., 2000] (see figure 2).

Because of inevitable model simplifications, the first method yields discrepancies between simulation results and reality. On the other hand, the second method resorts to uncontrolled and unrealistic parameters, and provides little physical insight of the system.

To overcome those drawbacks, we propose an original *joint utilization* of the two known complex-system simulation principles, i.e. of both the *a priori* and the *a posteriori* information, by making use of the measurement data in the simulation tool to enhance the simulation results (figure 3). We thus show that the combined utilization of physical models and measurement data can reduce the discrepancy between simulations and reality. To the best of our knowledge, this is the first work that propose this combination in the radio network context.

A MultiLayer Perceptron Artificial Neural Network with BackPropagation learning algorithm is used to find the stochastic mapping (transfer function) between the measurements and the simulations. The ANN is a two-phase learning system (*learning phase* and *validation/prediction*

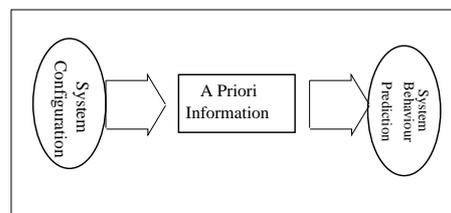


Fig. 1. System simulation using a priori information

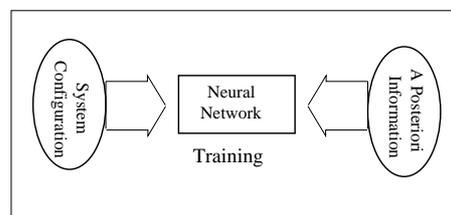


Fig. 2. System simulation using a posteriori information

*phase*). In the learning phase, two sets of data samples (the measurement data at hand and the simulation data) are presented to the MLP in order to learn the transfer function. In the validation/prediction phase, a validation data set is passed through the MLP to yield enhanced predictions.

Since we are interested in the statistical features of the variables rather than in the individual sample values, we propose that the ANN works on statistical distributions of the variables. This approach has been used for the face detection problem and has proven to yield good results [Waring and Liu, 2005].

To keep the computational complexity of the learning process tractable without losing the joint statistical properties of the variables, a preprocessing block of Independent Component Analysis (ICA) that allows us to work with individual 1-D distributions is used.

The proposed method is applied to a third generation (3G) cellular mobile network. The numerical results in the form of the analytical Kolmogorov-Smirnov test [Gabriel, 1978] show that the proposed approach is able to perform significant improvement on the training process results.

The remainder of this paper is structured as follows : In Section II we give a short overview of ICA theory. A detailed description of the proposed scheme is given in Section III. Section IV is devoted to a case study based on a 3G Radio Network Planning (RNP) tool followed by the

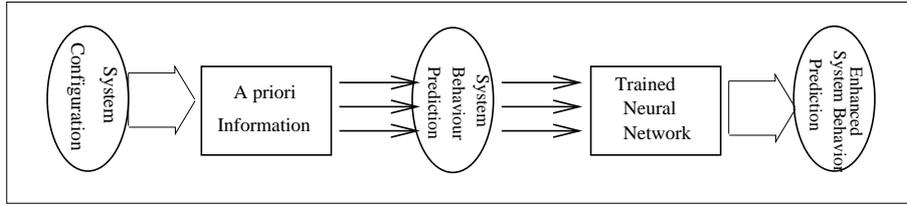


Fig. 3. System simulation using a priori and a posteriori information

results. Finally, Section V gives the concluding remarks.

## II. INDEPENDENT COMPONENT ANALYSIS

ICA is a linear technique that aims at extracting independent features from a data set. Unlike Principle Components Analysis (PCA) which simply decorrelates the data, ICA searches a linear, non-orthogonal coordinate system in the data-space whose components are independent across all statistical orders. ICA has been successfully applied to many problems such as the blind source separation problem [Amari et al., 1998], separation of artifacts in magnetoencephalograph data [Vigario et al., 1998], finding hidden factors in financial data [Kiviluoto and Oja, 1998], etc.

The statistical ICA model describes how the observed data  $x_1, x_2, \dots, x_n$  are generated by a process of mixing the independent *non-gaussian* components  $s_1, s_2, \dots, s_m$ . It can be written as  $\mathbf{x} = \mathbf{A}\mathbf{s}$  where  $\mathbf{x}$  is the observation vector composed of  $n$  linear mixtures  $x_i$  of  $m$  components  $s_i$ .

In the following, we make two assumptions for simplicity reasons :

1. The matrix  $\mathbf{A}$  is square
2. All independent components have identical distributions.

The research of the independent components is a two-step analysis. First, we estimate the mixing matrix  $\mathbf{A}$  using the observation vector  $\mathbf{x}$ . Then, we calculate the inverse of  $\mathbf{A}$ , denoted  $\mathbf{W}$ , which will allow to simply obtain the independent components :  $\mathbf{s} = \mathbf{W}\mathbf{x}$ .

To estimate one of the independent components we consider a linear combination of  $x_i$ ,  $\mathbf{y} = \mathbf{w}^T\mathbf{x}$  where  $\mathbf{w}$  is a vector to be determined. Thus  $\mathbf{y}$  is equal to one of the independent component if the  $\mathbf{w}^T$  are the rows of  $\mathbf{W}$ . By making the change of variables  $\mathbf{z} = \mathbf{A}^T\mathbf{w}$ , the vector  $\mathbf{y}$  is equal to  $\mathbf{z}^T\mathbf{s}$  and is a linear combination of  $s_i$ . According to the *central limit theorem*  $\mathbf{z}^T\mathbf{s}$  is more Gaussian than any of the  $s_i$  and becomes less Gaussian when it equals to one of the  $s_i$ . Therefore  $\mathbf{w}$  is a vector that maximizes the non-gaussianity of  $\mathbf{w}^T\mathbf{x}$ .

One popular measure of non-gaussianity is *negentropy* that is based on information theoretic quantity of entropy and is given by :  $J(\mathbf{y}) = H(\mathbf{y}_{Gauss}) - H(\mathbf{y})$  where  $\mathbf{y}_{Gauss}$  is a random Gaussian variable with the same covariance matrix as  $\mathbf{y}$  and  $H(\mathbf{y})$  is the entropy of  $\mathbf{y}$ . As a Gaussian variable has the largest entropy among all random variables of equal variance, maximizing non-gaussianity is equivalent to maximizing the negentropy.

One popular algorithm that performs ICA is FastICA, a

fixed point iteration scheme based on maximization of negentropy as a measure of non-gaussianity [Cover, 1991]. We have chosen to use FastICA in this study.

## III. PROPOSED SCHEME

Figures 4(a) and 4(b) depict the block diagrams of the learning and the prediction phases of the proposed scheme, respectively.

The training system proposed to learn the relation between simulations and measurements is based on Artificial Neural Networks (ANN) [Haykin, 1998]. This solution is chosen because of :

1. The relatively easy use of ANNs based on training-prediction cycles, and
2. Its powerful capacity to model extremely complex non linear functions.

In our proposed scheme we use one of the important class of ANNs : the MultiLayer Perceptrons (MLP) that consists of an input layer, one or more hidden layers, and an output layer. All these layers are formed by computational nodes. A node or a *perceptron* is the basic building block of a MLP and is composed of a linear combiner with adjustable weights or *free parameters*. The algorithm used to train the MLP is the very popular *error Back Propagation* (BP) algorithm which is the generalization of the famous gradient based Least-Mean-Square (LMS) algorithm.

As we are interested in statistical properties of the data rather than individual sample values, we propose that the neural network works on probability densities. Therefore, the data samples are first transformed to distributions (i.e. histograms) before entering the neural network and the training is done on histograms ( $\mathbf{h}_{sim}$  and  $\mathbf{h}_{mes}$ ). Each input/output of the neural network corresponds to a histogram bin. Thus, the number of inputs/outputs is determined by the number of histogram bins as well as the number of variables to predict.

For most of the cases, the system variables that are to be predicted are inter-correlated. Thus when we are interested in more than one variable, we have to use the joint distributions (i.e. multi-dimensional histograms) to conserve the joint statistical features and not to suffer from loss of mutual information between variables. However, the problem of using multi-dimensional distributions is that the number of inputs-outputs of the ANN increases exponentially, thus makes it rapidly untractable.

To overcome this problem we propose to use Independent Component Analysis (ICA) and work with independent components rather than with inter-correlated variables. Thus, the Neural Network can perform its learning and

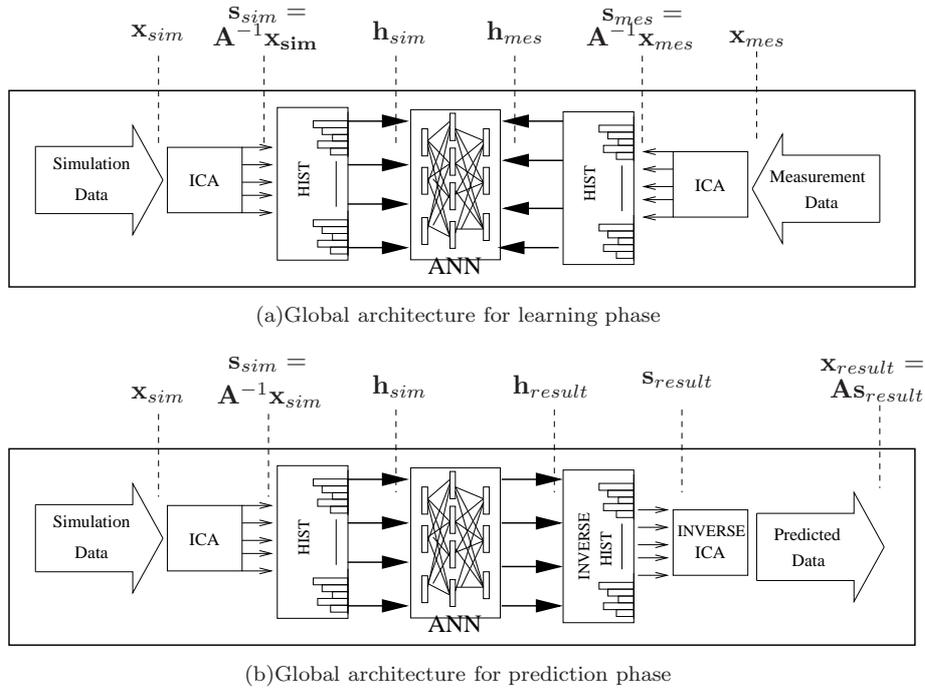


Fig. 4. Proposed scheme

generalization tasks on one-dimensional histograms of independent components without any loss of information. Another motivation for using ICA is the possibility to reduce the dimension of the data since the first step of the ICA algorithm is Principle Component Analysis (PCA) that makes possible the reduction of the number of significant components through Single Value Decomposition.

In the learning phase the following steps are followed :

1. The measurement data are first formatted : For example, each variable can be divided by its maximum value. This data scaling guarantees the non dominance of one variable with respect to another.
2. The formatted data are centered, i.e. the mean vector  $\mathbf{m} = E(\mathbf{x}_{mes})$  of the data is subtracted so to make  $\mathbf{x}_{mes}$  a zero mean variable.
3. The eigenvalues and eigenvectors of  $\mathbf{x}_{mes}$  are computed.
4. The measurement data  $\mathbf{x}_{mes}$  are linearly transformed so that a whitened vector is obtained. The whitening is done using the eigenvalue decomposition (EVD) of the covariance matrix  $E(\mathbf{x}_{mes}\mathbf{x}_{mes}^T) = \mathbf{E}\mathbf{D}\mathbf{E}^T$  where  $\mathbf{E}$  is the orthogonal matrix of eigenvectors of the covariance matrix and  $\mathbf{D}$  is the diagonal matrix of its eigenvalues.
5. The ICA matrix is computed using the FastICA algorithm.
6. Using matrix  $\mathbf{A}$ , we obtain a centred estimation of independent component  $\hat{\mathbf{s}}_{mes} = \mathbf{A}^{-1}(\mathbf{x}_{mes} - \mathbf{m})$
7. The mean vector of independent components given by  $\mathbf{A}^{-1}\mathbf{m}$  is added to the centred estimates so that  $\mathbf{s}_{mes} = \hat{\mathbf{s}}_{mes} + \mathbf{A}^{-1}\mathbf{m}$

We note here that there is only one computed ICA matrix : the one computed on the measurement data. This same matrix will be used for simulation data. Thus, we use the same coordinate system (same ICA matrix) and the existence of

a mapping between simulations and measurements to be learned is guaranteed.

Since the neural network has performed its learning on independent distributions, the outputs of the trained MLP are supposed to be independent histograms. In the prediction phase, the raw data ( $\mathbf{s}_{result}$ ) are obtained by calculating the inverse of the histogram transformation. Then the inverse of the computed ICA matrix,  $\mathbf{A}$ , is applied to data yielding dependent components, i.e  $\mathbf{x}_{result} = \mathbf{A}\mathbf{s}_{result}$ . Finally, the data are formatted according the scaling rules used in the learning phase.

In the chain shown in figure 4, the raw data samples (measurements and simulations in the learning phase, only simulations in the prediction phase) are first passed through ICA block yielding independent components, followed by a histogram block that forms one-dimensional histograms of each incoming data. Finally, the histogram bins form the inputs of the MLP for learning/prediction purposes. In the prediction phase, the outputs of the MLP are passed through an inverse chain to obtain the enhanced predictions.

#### IV. RESULTS

We have applied the proposed method to simulations of third generation (3G) Radio Networks using our Radio Network Planning (RNP) tool that performs Monte Carlo simulations of the Radio Access Network (RAN) according to predetermined traffic distributions and propagation model including correlated shadowing. It calculates the UpLink (UL) and DownLink (DL) transmission powers by taking into account phenomena such as mobility, macro diversity, admission control, load control, power control etc. At the output, it provides simulation results that consist of radio and quality indicators (UL/DL transmission powers and

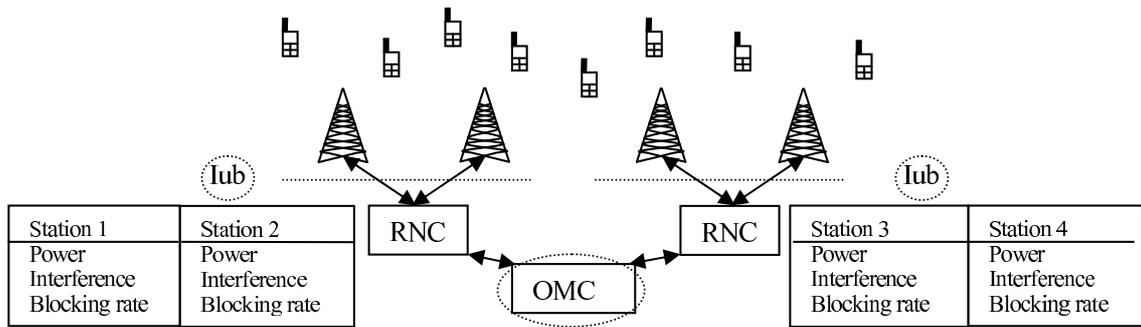


Fig. 5. 3G Radio Access Network and performance indicators

interferences, call dropping rate, call blocking rate, etc.). The quality indicators can be calculated for base stations, mobiles, or the whole network (voir figure 5).

The RNP tool has two operation modes : Static mode and dynamic mode. In the static mode, each Monte Carlo draw is an independent snapshot of the network whereas in the dynamic mode, subsequent draws are realizations of the network at successive time instants and therefore are correlated. In this work, we have confined our attention to the static mode.

The simulations yield two variables : Uplink Load (ULL) and Downlink Load (DLL) for each station in the network, for each snapshot. For test purposes, we have decided to create synthetically our measurement data with the RNP tool by modifying certain parameter(s). In other words, we conduct separate simulations with different parameters/configurations to obtain measurement data and simulation data. The measurement data are obtained by setting the target SIR\_UL (Signal-to-Interference Ratio in UL) parameter to -20 dB and the simulation data are obtained by setting the same parameter to -18 dB.

The proposed scheme is tested on a realistic UMTS network. The MLP used has one hidden layer having the same number of hidden neurons as the input and output layers. The method is applied to 2 stations and 2 indicators (ULL and DLL) with 20 bins in each histogram. For illustration purposes, we give the results by scatter plots. Each point in the scatter plot corresponds to a snapshot data sample. The vertical axis corresponds to the ULL and the horizontal axis to the DLL. The numerical results of the comparison are given by the 2-D Kolmogorov Smirnov test (KS-test) [Fasano and Franceschini, 1987] that determines the difference between two datasets. According to this test, two datasets are supposed to be coming from the same distribution if the value returned by the KS-Test is close to zero. If the two datasets are far from each other the KS-Test return a value near to 1.

Figure 6 shows the results of the learning phase where correlated raw data are directly used to train the MLP (we do not use ICA preprocessing). The black points correspond to measurements, dark grey points to simulations and the light grey points correspond to the outputs of the proposed scheme. Note that these are the results of the learning phase that are obtained by passing the simulation data set of the learning phase through the trained MLP. As shown in this figure the MLP cannot perform its learning

on raw data (the predicted-data distribution is far from the measurement distribution :  $KS\ distance = 0.23$ ).

To show the efficiency of the ICA processing module, we have also made a second test in which the MLP learning is performed with independent histograms rather than dependent ones. We have compared results using ICA with results when only PCA is used instead of ICA. Figure 7 shows that the MLP can perform its learning very well thanks to the ICA processing block. Furthermore comparing results given in figures 7 and 8, ICA ( $KS\ distance = 0.069$ ) is more efficient than PCA ( $KS\ distance = 0.12$ ). All these results are summarized in table I.

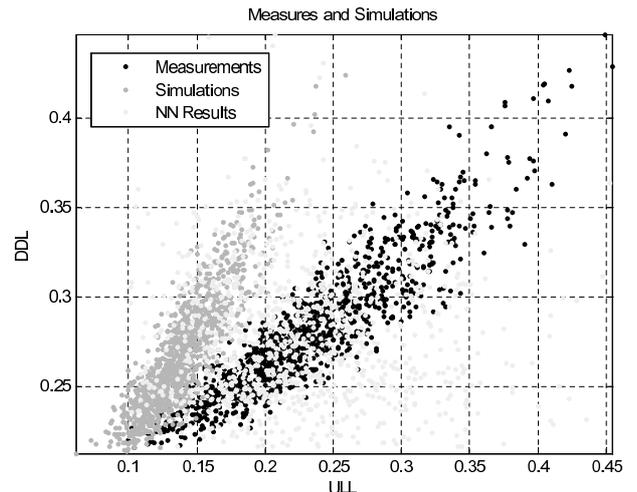


Fig. 6. Learning results without ICA use

## V. CONCLUSION

In this paper we propose a new method to enhance the predictions of a system simulator using measurements that makes use of both a priori (models) and a posteriori information (measurements). For this purpose we use a Multi-Layer Perceptron (MLP) Artificial Neural Network (ANN) with BackPropagation (BP) that finds an approximation of the stochastic transfer function between simulations and measurements. In the proposed scheme we propose to use Independent Component Analysis (ICA) as a preprocessing block that performs a coordinate transformation. This transformation will be used in the learning and prediction phases of the proposed scheme. With such a transformation we are able to work with multidimensional variables

TABLE I: Comparison of 2-D Kolmogorov Smirnov Distance

	Without ICA use	With ICA use	With PCA use
Mesurement-Simulation	0.737	0.737	0.737
Mesurement-NN Result	0.23	0.069	0.12

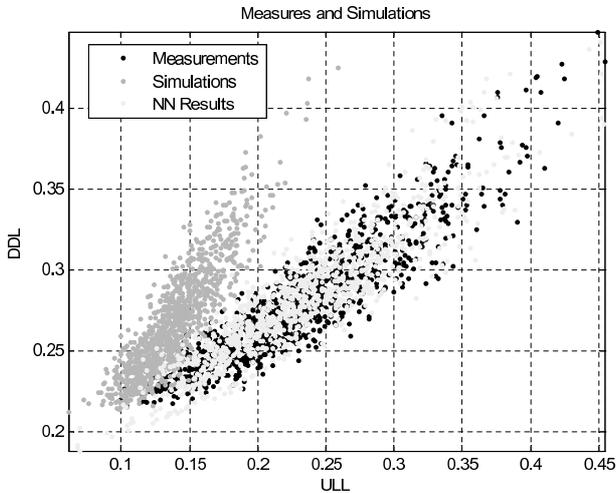


Fig. 7. Learning results with ICA use

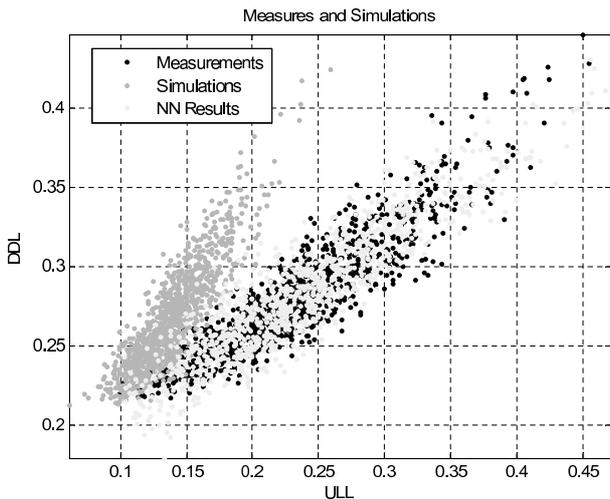


Fig. 8. Learning results with PCA use

#### ACKNOWLEDGEMENT

Authors would like to thank M. Christophe Gay for his very valuable help on computing skills.

#### REFERENCES

- Amari, S., Cichocki, A., and Yang, H. (1998). A new learning algorithm for blind source separation. In *Proc. Advances in Neural Information Processing Systems*, volume 8, pages 757–763. MIT Press.
- Andrea, N., Cecchetti, C., and Lipparwi, A. (2000). Fast prediction of the performance of wireless links by simulation trained neural network. *Proc. IEEE MTT-S Digest 2000*, pages 429–432.
- Centeno, M. and Reyes, M. (1998). So you have your model : what to do next ? a tutorial on simulation output analysis. *Simulation Conference Proceedings*, 1 :23–29.
- Cover, T. M. (1991). *Elements of information theory*. John Wiley and Sons.
- Fasano, G. and Franceschini, A. (1987). A multidimensional version of the kolmogorov-smirnov test. *Royal Astronomical Society*, 255 :155–170.
- Gabriel, K. R. (1978). A simple method of multiple comparisons of means. *Journal of the American Statistical Association*, 73 :364.
- Haykin, S. (1998). *Neural Networks : A comprehensive foundation*. Prentice Hall, second edition.
- Kiviluoto, K. and Oja, E. (1998). Independent component analysis for parallel financial time series. In *Proc. International Conference on Neural Information Processing (ICONIP'98)*, volume 2, pages 895–898, Tokyo, Japan.
- Vigario, R., Jousmaki, V., Hamalainen, M., and Oja, E. (1998). Independent component analysis for identification of artifacts in magnetoencephalographic recordings. In *Proc. Advances in Neural Information Processing Systems*, volume 10, pages 229–235. MIT Press.
- Waring, C. and Liu, X. (2005). Face detection using spectral histograms and svms. *IEEE Transactions on Systems, Man, and Cybernetics, Part B : Cybernetics*, pages 467– 476.

without losing their mutual statistical information and ensuring a tractable computational complexity.

We have applied the proposed method to simulation results of a 3G Radio Network Planning tool where we have used a synthetically generated measurement data (the synthetic measurement data is also generated by the RNP tool by modifying certain parameters). The results show that the proposed scheme using ICA is able to learn and predict the stochastic transfer function between simulations and measurements. We have show also that ICA performs better than PCA. Future work will include testing this method on real measurement data, and generalizing its use in our RNP tool.