# PROPERTIES OF PREDICTOR BASED ON RELATIVE NEIGHBORHOOD GRAPH LOCALIZED FIR FILTERS.

*John Aasted Sørensen*

Electronics Institute, Build. 349, Technical University of Denmark, 2800 Lyngby, Denmark

## ABSTRACT

A time signal prediction algorithm based on Relative Neighborhood Graph (RNG) localized FIR filters is defined. The RNG connects two nodes, of input space dimension $D$, if their lune does not contain any other node. The FIR filters associated with the nodes, are used for local approximation of the training vectors belonging to the lunes formed by the nodes. The predictor training is carried out by iteration through 3 stages: Initialization of the RNG of the training signal by vector quantization, LS estimation of the FIR filters localized in the input space by RNG nodes and adaptation of the RNG nodes by equalizing the LS approximation error among the lunes formed by the nodes of the RNG.

The training properties of the predictor is exemplified on a burst signal and characterized by the normalized mean square error (NMSE) and the mean valence of the RNG nodes through the adaptation.

## 1. Introduction.

A time signal predictor based on the Relative Neighborhood Graph (RNG), [1] used for localizing finite impulse response (FIR) filters in the input space of dimension $D$ of a training signal, is proposed.

The predictor is trained during 3 stages:

**Stage 1:** Initialize the RNG which quantize the input space of the training signal.

**Stage 2:** For fixed RNG, estimate the localized FIR filters, associated the nodes of the RNG.

**Stage 3:** **If** the prediction error of the training signal is sufficiently low **then** terminate training **else** adapt the RNG to the training signal and continue from Stage 2.

It is assumed that a training signal $x_n$, $n = 1, \ldots, N$ and a dimension $D$ of the input space is given. This defines the input signal vector $\mathbf{x}_n^T = (x_n, \ldots, x_{n-D+1})$ and an augmented input vector $\mathbf{z}_n^T = (\mathbf{x}_n^T, 1)$. From this, the training signal data matrix becomes $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$.

## 2. The Relative Neighborhood Graph.

The RNG of the column vectors of the matrix $\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_R]$ where $\mathbf{p}_i^T = (p_{1,i}, \ldots, p_{D,i})$ associate a node number $i$ to the column vector number $i$.

A pair of RNG nodes $i$ and $j$ are connected, if the lune $\Lambda_{i,j}$ of the corresponding vectors $\mathbf{p}_i$ and $\mathbf{p}_j$ does not contain any other column vector from $\mathbf{P}$.

If the sphere with center in $\mathbf{x}$ and radius $r$ is denoted

$$B(\mathbf{x}, r) = \{\mathbf{y} \mid d(\mathbf{x}, \mathbf{y}) \le r\} \qquad (1)$$

where $d(\mathbf{x}, \mathbf{y})$ is the distance between $\mathbf{x}$ and $\mathbf{y}$, then the lune $\Lambda_{i,j}$ of $\mathbf{p}_i$ and $\mathbf{p}_j$ is determined by

$$\Lambda_{i,j} = B(\mathbf{p}_i, d(\mathbf{p}_i, \mathbf{p}_j)) \cap B(\mathbf{p}_j, d(\mathbf{p}_i, \mathbf{p}_j)) \qquad (2)$$

or by

$$\Lambda_{i,j} = \{\mathbf{x} \mid \max(d(\mathbf{p}_i, \mathbf{x}), d(\mathbf{x}, \mathbf{p}_j)) \le d(\mathbf{p}_i, \mathbf{p}_j)\}. \qquad (3)$$

The RNG incidence matrix $\mathbf{C} = [C_{i,j}]$, $i, j = 1, \ldots, R$, where the element $C_{i,j} = 1$ if the RNG nodes $i$ and $j$ are connected, otherwise $C_{i,j} = 0$. The $R$ column sums of $\mathbf{C}$ represents the valences of the respective nodes of the RNG. A 3-node RNG with $D = 2$, $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3]$, valences 1, 2 and 1 and the Euclidean distance measure, is exemplified in Fig. 1. Here $\mathbf{x}_j$ belongs to the lune of node 1 and 2 and $\mathbf{x}_i$ to the lune of node 2 and 3, furthermore $\mathbf{x}_i$ belongs to the intersection of the two lunes. The $\mathbf{x}_k$ does not belong to any lune.

## 3. Training Algorithm.

The predictor training algorithm, [4] requires the following 3 stages:

**Stage 1: Initialization of RNG.**

$$[\mathbf{P}, \mathbf{C}] = RNG(\mathbf{X}). \qquad (4)$$

The method of initializating the RNG of $\mathbf{X}$, such that all column vectors of $\mathbf{X}$ belongs to at least one lune of the RNG is as follows: Select randomly a few seed vectors from $\mathbf{X}$ and use them as RNG nodes. Then

0-7803-2431-5/95 $4.00 © 1995 IEEE

associate the vectors of $\mathbf{X}$ not member of any lune, to their nearest node. Now for each node, select among the associated vectors, that vector with the largest distance from that node and include it as a new RNG node. Generate the new RNG and repeat this initialization operation, until all columns of $\mathbf{X}$ are members of at least one RNG lune.

**Stage 2: Estimation of Local Filters.**
To each node $i$ of the RNG is associated a FIR filter $\mathbf{w}_i^T = (w_{1,i}, w_{2,i}, \ldots, w_{D,i}, w_{D+1,i})$, where the element $w_{D+1,i}$ is the bias term of the filter. All filters are then represented by

$$\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_R]. \tag{5}$$

The basis of $\mathbf{W}$ estimation, is the association of each training vector to the nodes of the RNG. Let the association matrix

$$\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_N] \tag{6}$$

where $\mathbf{a}_n^T = (a_{1,n}, \ldots, a_{R,n})$ and $a_{i,n}$ is the fraction of times the RNG node $i$ takes part in forming lunes to which $\mathbf{x}_n$ belongs. From this the $\sum_{i=1}^R a_{i,n} = 1$.
Depending on how the input signal to the local filters are generated, 2 different local filters $\mathbf{W}$ and $\mathbf{V}$ can be defined:

In the first filter, $\mathbf{x}_n$ is projected directly on the local filters $\mathbf{W}$, which leads to the predictor:

$$\hat{x}_{n+1} = \sum_{i=1}^{R} a_{i,n}\mathbf{w}_i^T \mathbf{z}_n \tag{7}$$

From this is obtained

$$\hat{x}_{n+1} = \mathbf{a}_n^T \mathbf{W}^T \mathbf{z}_n = (\mathbf{W}\mathbf{a}_n)^T \mathbf{z}_n \tag{8}$$

which leads to

$$\hat{x}_{n+1} = \mathbf{W}_n^T \mathbf{z}_n \tag{9}$$

where $\mathbf{W}_n$ are the time varying predictor coefficients.

In the second filter, $\mathbf{x}_n - \mathbf{p}_i$, $i = 1, \ldots, R$ are projected on the local filters $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_R]$, where $\mathbf{v}_j^T = [v_{1,j}, \ldots, v_{D+1,j}]$. Using $\mathbf{Y}_n = [\mathbf{y}_{1,n}, \ldots, \mathbf{y}_{R,n}]$, where $\mathbf{y}_{i,n}^T = [\mathbf{x}_n^T - \mathbf{p}_i^T, 1]$ leads to the predictor:

$$\hat{x}_{n+1} = \sum_{i=1}^{R} a_{i,n}\mathbf{v}_i^T \mathbf{y}_{i,n}. \tag{10}$$

Rewriting leads to

$$\hat{x}_{n+1} = trace(\mathbf{V}^T \mathbf{Y}_n diag(\mathbf{a}_n)) \tag{11}$$

and

$$\hat{x}_{n+1} = trace(diag(\mathbf{a}_n)\mathbf{V}^T \mathbf{Y}_n) = trace(\mathbf{V}_n^T \mathbf{Y}_n) \tag{12}$$

where $\mathbf{V}_n$ are the time varying predictor coefficients.

For a fixed RNG the LS-estimator of $\mathbf{W}$ becomes:

$$\mathbf{R}\mathbf{w} = \mathbf{r} \tag{13}$$

where $\mathbf{R} = [\mathbf{R}_{i,j}]$, $i, j = 1, \ldots, R$, $\mathbf{w}^T = [\mathbf{w}_1^T, \ldots, \mathbf{w}_R^T]$ and $\mathbf{r}^T = [\mathbf{r}_1^T, \ldots, \mathbf{r}_R^T]$.
The matrices of Eq. 13 are found as follows:

$$\mathbf{R}_{i,j} = \sum_{n=1}^{N} (a_{i,n-1}\mathbf{z}_{n-1})(a_{j,n-1}\mathbf{z}_{n-1})^T \tag{14}$$

which is the correlation matrix between the augmented input signal $\mathbf{z}_{n-1}$, weighted by the fraction of times RNG node $i$ and $j$ takes part in forming lunes to which $\mathbf{z}_{n-1}$ belongs.

$$\mathbf{r}_i = \sum_{n=1}^{N} a_{i,n-1} x_n \mathbf{z}_{n-1} \tag{15}$$

which is the correlation vector between $x_n$ being predicted, and the input signal $\mathbf{z}_{n-1}$ weighted by the fraction of times, the RNG node $i$ takes part i forming the lunes to which $\mathbf{z}_{n-1}$ belongs.

Using Eq. 13 in the case of a fixed RNG, the LS-estimator of $\mathbf{V}$ becomes:

$$\mathbf{R}_{i,j} = \sum_{n=1}^{N} (a_{i,n-1}\mathbf{y}_{i,n-1})(a_{j,n-1}\mathbf{y}_{j,n-1})^T \tag{16}$$

which is the weighted correlation matrix between the augmented input signals $\mathbf{y}_{i,n-1}$ of RNG filter number $i$ and $\mathbf{y}_{j,n-1}$ of node $j$.
The right hand side becomes

$$\mathbf{r}_i = \sum_{n=1}^{N} a_{i,n-1} x_n \mathbf{y}_{i,n-1} \tag{17}$$

which is the weighted correlation vector between $x_n$ being predicted, and the input signal $\mathbf{y}_{i,n-1}$ of filter number $i$.

**Stage 3: Adaptation of the RNG to the training signal.**
The training signal squared error matrix $\mathbf{E} = [E_{i,j}]$, $i, j = 1, \ldots, R$ of the RNG is then determined as follows: Let $e_n = x_n - \hat{x}_n$, $n = 1, \ldots, N$ be the training signal estimation error at time $n$. Then the squared

error $e_n^2$ is divided equally among the lunes from which $\hat{x}_n$ is generated, [5]. If the RNG node $i$ and node $j$ forms a lune ($C_{i,j} = 1$), then $E_{i,j}$ is the sum of squared errors of the training signal associated that lune.
The adaptation rule of the nodes
$\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_i, \ldots, \mathbf{p}_R]$ then becomes:
If the valence of node $i$ is $> 1$ then adapt $\mathbf{p}_i$ in a direction such that the lune with the minimum squared error, which $\mathbf{p}_i$ takes part in forming, is enlarged in size, and the lunes with larger squared errors are decreased in size. This leads to the following adaptation of $\mathbf{P}$:

$$\mathbf{P}_{new} = \mathbf{P}_{old} + \Delta\mathbf{P} \qquad (18)$$

where

$$\Delta\mathbf{P} = [\Delta\mathbf{p}_1, \ldots, \Delta\mathbf{p}_i, \ldots, \Delta\mathbf{p}_R] \qquad (19)$$

If the above minimum squared error is denoted $minE_i$ then the adaptation of node $i$ becomes:

$$\Delta\mathbf{p}_i = \sum_{j=1}^{R} \frac{E_{i,j} - minE_i}{E_{i,j}} C_{i,j}(\mathbf{p}_j - \mathbf{p}_i)\mu \qquad (20)$$

where $\mu$ is the adaptation constant.

## 4. Investigation of the training algorithm properties.

The RNG training is carried out using the predictor (7) in the case of a burst signal generated from the Subba Rao model, [2], [4]. The signal is shown in Fig. 2. In Fig. 3 and Fig. 5 are shown the $NMSE$ depending of the number of iterations through the training signal. In Fig. 4 and Fig. 6 the corresponding mean valence of the RNG nodes are shown.
The initializing parameters of the training are shown in the following table:

| $D$ | $\mu$ | $R_{init.}$ | $R_{adapted}$ | $NMSE$ and $valence$ |
|---|---|---|---|---|
| 2 | 0.01 | 3 | 36 | Fig. 3 and Fig. 4 |
| 3 | 0.01 | 3 | 34 | Fig. 5 and Fig. 6 |

From the training experiments it is seen that the two cases obtain approximately the same $NMSE$ values; but the mean $RNG$ valence for $D = 2$ becomes larger than the mean $RNG$ valence for $D = 3$, to compensate the reduced local approximation capability in the case of $D = 2$.

## References.

[1 ] Jerzy W. Jaromczyk, Godfried T. Tousssaint, *Relative Neighborhood Graphs and Their Relatives*, Proceedings of IEEE, Vol. 80, No. 9, September 1992.

[2 ] M.B. Priestly, *Non-linear and Non-stationary Time series Analysis*. Academic Press, 1988.

[3 ] *Time Series Prediction*,
Andreas S. Weigend, Neil A. Gershenfeld, Eds.
Proceedings Volume XV, Santa Fe Institute
Addison-Wesley Publishing Company, 1994.

[4 ] John Aasted Sørensen, *Time Signal Filtering by Relative Neighborhood Graph Localized Linear Approximation*. Proc. 1994 IEEE Workshop on Neural Networks for Signal Processing.

[5 ] John Aasted Sørensen, *Quantized, Piecewise Linear Filter Network*. Proc. 1993 IEEE Workshop on Neural Networks for Signal Processing.

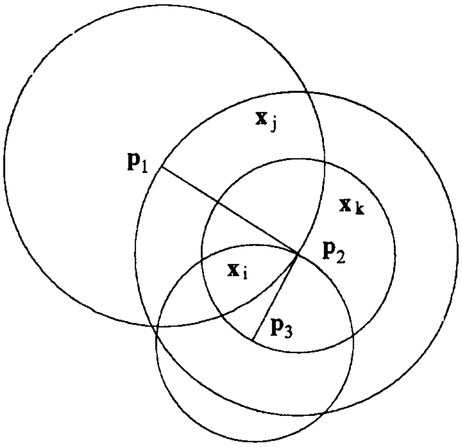

Figure 1: Example 3-node RNG.

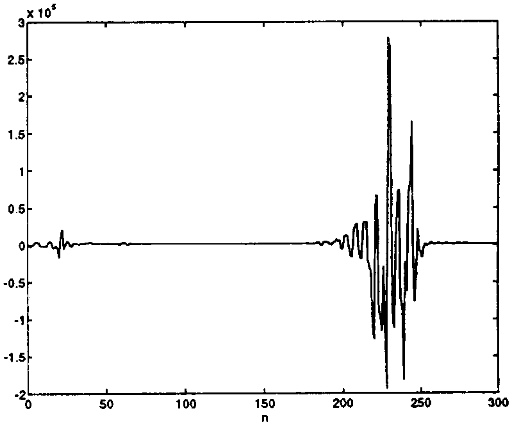

Figure 2: Subba Rao training signal.

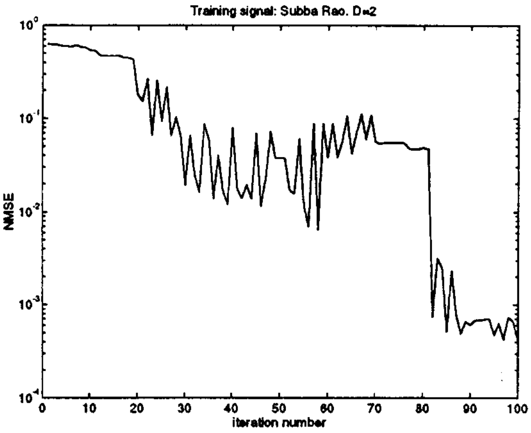

Figure 3: NMSE for training on Subba Rao with D=2.

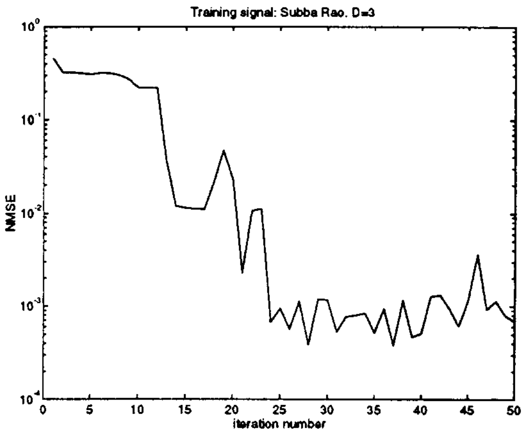

Figure 5: NMSE for training on Subba Rao with D=3.

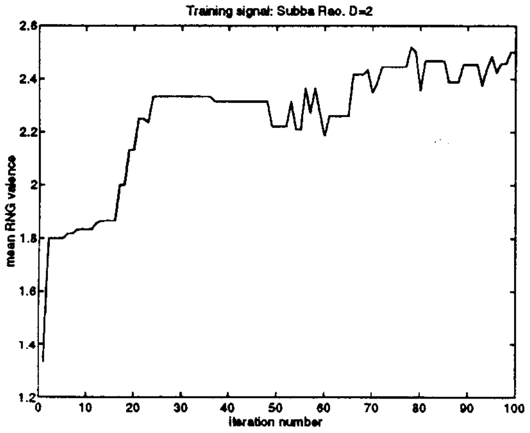

Figure 4: Mean RNG valence for D=2.

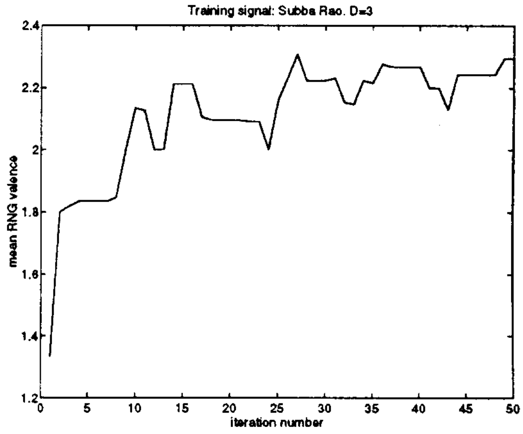

Figure 6: Mean RNG valence for D=3.