Global Journal of Computer Science and Technology: G Interdisciplinary
Volume 17 Issue 3 Version 1.0 Year 2017
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

From Forward Prediction Error and Backward Prediction Error to Orthogonal Data in Space(Lattice Predictor) and the Origin of a System to Pick up Another

By Dr. Ziad Sobih
Northeastern University

Abstract- In this paper, we will develop another class of linear filter which involve order update and time update. These filters have the important fact of order update. We will show a computationally efficient modular lattice-like architecture. This lead to a filter with computational complexity linear with the order which is the length.

The design of order recursive adaptive filter can take two approaches.

1. Stochastic [16] gradient approach. This is Wiener theory.

2. Least squares approach. This is Kalman filter theory.

The second approach is code demanding. We will start with the first approach.

Keywords: wiener theory, prediction, filters, stochastic gradient, learning and lattice filter.

GJCST-G Classification: J.2

FROMFORWARDPREDICTIONERRORANDBACKWARDPREDICTIONERRORTOORTHOGONALDATAINSPACELATTICEPREDICTORANDTHEORIGINOFASYSTEMTOPICKUPANOTHER

Strictly as per the compliance and regulations of:

From Forward Prediction Error and Backward Prediction Error to Orthogonal Data in Space(Lattice Predictor) and the Origin of a System to Pick up Another

Dr. Ziad Sobih

Abstract- In this paper, we will develop another class of linear filter which involve order update and time update. These filters have the important fact of order update. We will show a computationally efficient modular lattice-like architecture. This lead to a filter with computational complexity linear with the order which is the length.

The design of order recursive adaptive filter can take two approaches.

1. Stochastic [16] gradient approach. This is Wiener theory.
2. Least squares approach. This is Kalman filter theory.

The second approach is code demanding. We will start with the first approach.

Keywords: wiener theory, prediction, filters, stochastic gradient, learning and lattice filter.

I. Introduction

The adaptive gradient lattice (GAL) filter is due to Griffiths (1977) and may be viewed as a natural extension of least mean square as they both use stochastic gradient [16] approach. First, we derive the recursive formula for order update then we find the updates for the desired response.

II. Multistage Lattice Predictor [18]

Figure 2 is a single stage lattice predictor. The input and output are characterized by a single parameter km. We assume that the input is wide sense stationary. To find km, we start with the cost function.

$$J_{fb,m}=\frac{1}{2}\ E[|f_m(n)|^2+|b_m(n)|^2] \qquad (1)$$

Where fm(n) is the forward prediction error and bm(n) is the backward prediction error and E is the expected value. The relation for the lattice from stage m-1 to m is

$$f_m(n)=f_{m-1}(n)+k_m^{*}b_{m-1}(n-1) \quad (2) \qquad (2)$$

$$b_m(n)\ =\ b_{m-1}(n-1)\ +\ k_m f_{m-1}(n-1) \qquad (3)$$

Using equations 1 and 2 and 3 we will have for

$$J_{fb,\,m}=\frac{1}{2}(E[|f_{m-1}(n)|^2]+E[|b_{m-1}(n-1)|^2])(1+|k_m|^2)$$
$$+k_m E[f_{m-1}(n)\ b^{*}_{m-1}(n-1)]$$
$$+k_m^{*}[b_{m-1}(n-1)\ f^{*}_{m-1}(n)] \qquad (4)$$

Author: Northeastern University, Boston, MA.
e-mail: Sobih84@gmail.com

This is a max-min problem. We want to find the min j as km change. Differentiating

$$\frac{\partial J}{\partial k}=k_m(E[|f_{m-1}(n)|^2]+E[|b_{m-1}(n-1)|^2])$$
$$+2E[b_{m-1}(n-1)f^{*}_{m-1}(n)] \qquad (5)$$

Equating to zero we find that the optimum value of km to make j minimum.

$$k_{m,o}=-2\,E\,[b_{m-1}(n-1)\ f^{*}_{m-1}(n)]$$
$$/\ (E[|f_{m-1}(n)|^2+|b_{m-1}(n-1)|^2]) \qquad (6)$$

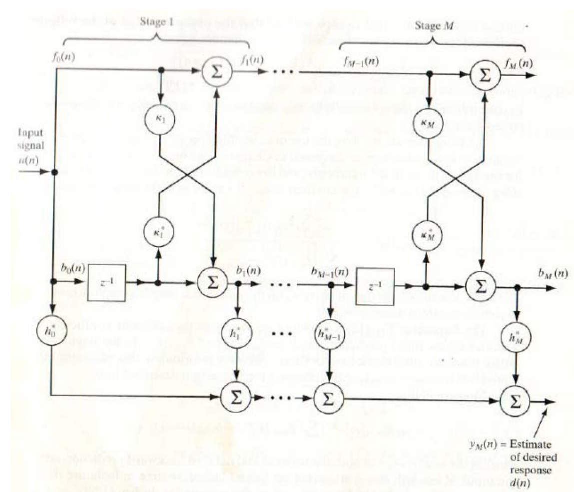

Figure 1: Lattice filter

This is Burg formula (1968).

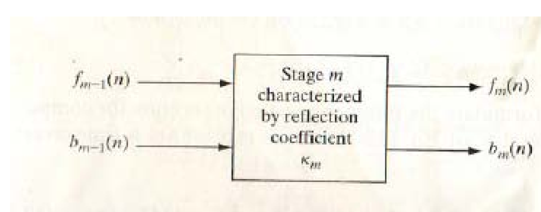

Figure 2: Block diagram

This formula assumes that the process is ergodic. This means we can use time averages. We get for the reflection coefficient km for the m stage in the lattice predictor

$$k_m(n)= - 2 \sum_{i=1}^{n} [b_{m-1}(i-1)\, f^{*}_{m-1}(i)]$$

$$/ \left(\sum_{i=1}^{n} [|f_{m-1}(i)|^2+|b_{m-1}(i-1)|^2]\right) \qquad (7)$$

It is clear that the estimate is data dependent.

Equation 7 is a block estimator for the reflection coefficient km. It is time now to find a recursive formula to update km.

First, we find

$$E_{m-1}(n)=\sum_{i=1}^{n} [|f_{m-1}(i)|^2+ | b_{m-1}(i-1)|^2] \qquad (8)$$

This is the total energy of the forward and delayed backward error at the input of the m stage. Doing some math, we will have the recursive formula.

$$E_{m-1}(n)=\sum_{i=1}^{n-1} [|f_{m-1}(i)|^2+ | b_{m-1}(i-1)|^2]$$

$$+[| b_{m-1} (n-1)|^2+| f^{*}_{m-1}(n)|^2]$$

$$=E_{m-1}(n-1)+| b_{m-1} (n-1)|^2+| f_{m-1}(n)|^2] \qquad (9)$$

At this point, we need a recursive formula for equation 7 and we will start by writing the top as

$$\sum_{i=1}^{n} [b_{m-1}(i-1)\, f^{*}_{m-1}(i)]$$

$$= \sum_{i=1}^{n-1} [b_{m-1} (i-1)\, f^{*}_{m-1}(i)] + [b_{m-1} (n-1)\, f^{*}_{m-1}(n)] \qquad (10)$$

Substituting equations 9 and 10 into 7, we will find that

$$k_m(n) =\sum_{i=1}^{n-1} [b_{m-1} (i-1)\, f^{*}_{m-1}(i)] + [b_{m-1} (n-1)\, f^{*}_{m-1}(n)]$$

$$/ \, E_{m-1}(n-1)+ | b_{m-1} (n-1)|^2+| f_{m-1}(n)|^2] \qquad (11)$$

Equation 11 is not a pure recursive form, so we need to do some more steps.

First use km(n-1) in place km in equation 2 and 3 and write them as

$$f_m(n) = f_{m-1}(n) + k_m(n-1)^{*} b_{m-1}(n-1) \qquad (12)$$

Second use equation 12 and 13 with 9 to write

$$b_m(n) = b_{m-1} (n - 1) + k_m(n-1)\, f_{m-1}(n - 1) \qquad (13)$$

$$2b_{m-1}(n-1)f^{*}_{m-1}(n)=b_{m-1}(n-1)f^{*}_{m-1} +f^{*}_{m-1}(n)b_{m-1}(n-1)$$

$$=b_{m-1}(n-1)(f_m(n)-k^{*}_{em}(n-1)b_{m-1}(n-1))^{*}$$

$$+f^{*}_{m-1}(n)(b_m(n)-k_{em}(n-1)f_{m-1}(n))$$

$$=-k_{em}(n-1)(|f_{m-1}(n)|^2+|b_{m-1}(n-1)|^2)$$

$$+(f^{*}_{m-1}(n)b_m(n)+b_{m-1}(n-1)f^{*}_m(n))$$

$$= - k_{em}(n-1)E_{m-1}(n)+k_{em}(n-1)E_{m-1}(n-1)$$

$$+(f^{*}_{m-1}(n)b_m(n)+b_{m-1}(n-1)f^{*}_m(n))$$

Then we use equation 7 for (n-1) to write equation 11 as

$$2\sum_{i=1}^{n-1} b_{m-1}(i-1)f_{m-1}^{*}(i)+2b_{m-1}(n-1)f^{*}_{m-1}(n)$$

$$=k_{em}(n-1)E_{m-1}(n-1)-k_{em}(n-1)E_{m-1}(n)+k_{em}(n-1)E_{m-1}(n-1)$$

$$+(f^{*}_{m-1}(n)b_m(n)+b_{m-1}(n-1)f_m^{*}(n))$$

$$= -k_{em}(n-1)E_{m-1}(n)+(f_{m-1}^{*}(n)b_m(n)+b_{m-1}(n-1)f^{*}_m(n))$$

This mean

$$k_{em}(n)=k_{em}(n-1)-(f^{*}_{m-1}(n)b_m(n)+b_{m-1}(n-1)f^{*}_m(n))/E_{m-1}(n)$$

$$m=1,2,\ldots\ldots,M. \qquad (14)$$

At this point, we will make two modification to equations 9 and 14.

1. We will introduce a step size parameter to control the adjustment.

$$k_{em}(n)=k_{em}(n-1)-[\mu_e/E_{m-1}(n)](f^{*}_{m-1}(n)b_m(n)$$

$$+b_{m-1}(n-1)f^{*}_m(n))$$

$$M=1,2, \ldots\ldots,M. \qquad (15)$$

2. We introduce an averaging filter to the energy estimator

$$E_{m-1}(n)=\beta E_{m-1}(n-1)+(1-\beta)(|f_{m-1}(n)|^2+|b_{m-1}(n-1)|^2) \qquad (16)$$

Equation 16 take the fact that we are dealing with nonstationary environment, and we have statistical variation. This will equip the estimator with memory were the present value and immediate past is used.

III. Desired Response Estimator [14]

Let us say we want a desired response d(n). we consider the structure shown in figure 3 which is part of figure 1. We have the input vector bm(n) and the parameters of the filter hm(n) which will converge with time to give the desired response.

For the estimation of the vector h we use the stochastic gradient approach. We find that the order update for the desired response d(n) is

$$y_m(n)= \sum_{k=0}^{m} h_{ek}^{*}(n)b_k(n)$$

$$= \sum_{k=0}^{m-1} h_{ek}^{*}(n)b_k(n) + h_{ek}^{*}(n)\, b_k(n) \qquad (17)$$

$$= y_{m-1}(n) + h_{ek}^{*}(n)\, b_k(n)$$

The error is

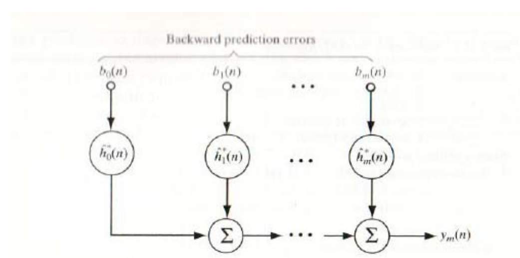

Figure 3: The coefficients h

$$e_m(n)=d(n)-y_m(n) \qquad (18)$$

The time update for the mth coefficient of figure 3 is

$$h_{em}(n+1)=h_{em}(n)+[\mu/||\mathbf{b_m}(n)||^2]b_m(n)e^{*}_m(n) \qquad (19)$$

The squared Euclidean norm is defined as

$$||\mathbf{b_m}(n)||^2 =\sum_{k=0}^{m} |b_k(n)|^2$$

$$=|b_m(n)|^2 + \sum_{k=0}^{m-1} |b_k(n)|^2$$

$$= ||\mathbf{b_m}(n)||^2 + |b_k(n)|^2 \qquad (20)$$

IV. Adaptive Forward Linear Prediction [17]

Conceder the 4th order filter in figure 4 at time n. The forward prediction error is

$$f_m(i)=u(i)-\mathbf{w}_{ef,m}^{H}(\mathbf{n})\mathbf{u}_m(\mathbf{i-1}) \tag{21}$$

The forward prediction problem is to find u(i) at time i from the vector u(i-1)…………u(i-m) using the filter in figure 4 of the weight vector wm1(n)…………..wmm(n).

We refer to fm(i) as the forward a posteriori prediction error, since its value is based on the current weight vector wfm(n). We defined forward a priori prediction error as

$$\mathbf{u(i-1)}=[u(i-1), u(i-2),\ldots\ldots\ldots,u(i-m)]^T$$

$$\mathbf{w(n)}=[w_{f,m,1}, w_{f,m,2}(n),\ldots\ldots\ldots,w_{f,m,m}(n)]^T \tag{22}$$

The update formula for the weights vector for the forward predictor is

$$\eta_m(i)=u(i)-\mathbf{w}_{ef,m}(\mathbf{n-1})\,\mathbf{u}_m(\mathbf{i-1})$$

$$I = 1, 2, \ldots\ldots.,n. \tag{23}$$

k is the gain vector defined by

$$\mathbf{k}_m(\mathbf{n-1})=\boldsymbol{\phi}_m^{-1}(\mathbf{n-1})\mathbf{u}_m(\mathbf{n-1}) \tag{24}$$

In equation 24 we have the inverse of the correlation matrix defined.

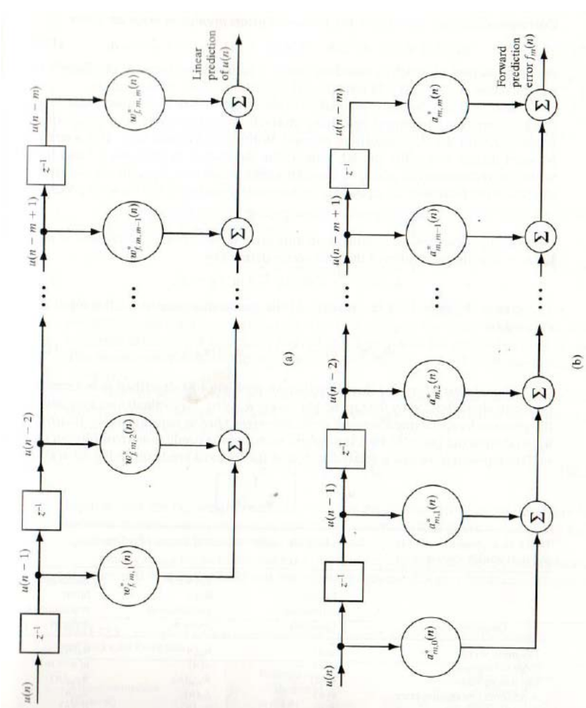

Figure 4: Forward prediction

$$\boldsymbol{\Phi}_m(\mathbf{n-1})=\sum_{i=1}^{n-1} \lambda^{n-1-i}\mathbf{u}_m(\mathbf{i})\mathbf{u}_m^{H}(\mathbf{i}) \tag{25}$$

At this point, we have described the adaptive filter forward prediction problem and using the weight vector w,f,m(n). Also, the forward prediction error problem is important and we are going to approach the solution using the knowledge we have so far. Let us say we have am(n) were [15].

$$\mathbf{a}_m(\mathbf{n}) = \begin{bmatrix} 1 \\ -\boldsymbol{w} \end{bmatrix} \tag{26}$$

Table 2: Notation

Quantity	Linear estimation (general)	Forward linear prediction of order m	Backward linear prediction of order m
Tap-input vector	$\mathbf{u}(n)$	$\mathbf{u}_m(n-1)$	$\mathbf{u}_m(n)$
Desired response	$d(n)$	$u(n)$	$u(n-m)$
Tap-weight vector	$\hat{\mathbf{w}}(n)$	$\hat{\mathbf{w}}_{f,m}(n)$	$\hat{\mathbf{w}}_{b,m}(n)$
A posteriori estimation error	$e(n)$	$f_m(n)$	$b_m(n)$
A priori estimation error	$\xi(n)$	$\eta_m(n)$	$\beta_m(n)$
Gain vector	$\mathbf{k}(n)$	$\mathbf{k}_m(n-1)$	$\mathbf{k}_m(n)$
Minimum value of sum of weighted error squares	$\mathscr{E}_{min}(n)$	$\mathscr{F}_m(n)$	$\mathscr{B}_m(n)$

Where the first element of the vector am(n) is one. The forward a posteriori prediction error and the forward a priori prediction error

$$f_m(i) = \mathbf{a}_m^{H}(\mathbf{n})\mathbf{u}_{m+1}(\mathbf{i})$$

$$i = i=1, 2,\ldots\ldots, n, \tag{27}$$

And

$$\eta = \mathbf{a}_m^{H}(\mathbf{n-1})\mathbf{u}_m(\mathbf{i})$$

$$i=1, 2, \ldots\ldots., n, \tag{28}$$

The input vector of size m+1 is the following,

$$\mathbf{u}_{m+1}(\mathbf{i}) = \begin{bmatrix} u(i) \\ \boldsymbol{u(i-1)} \end{bmatrix}$$

Because of orthogonality we have the condition,

$$\sum_{i=1}^{n} \lambda^{n-1}\mathbf{u(i-1)}f_m^{*}(i) = \mathbf{0} \tag{29}$$

The weight vector w f,m(n) can also be found by minimizing the sum

$$F_m(n) = \sum_{i=1}^{n} \lambda^{n-1}|f_m(i)|^2 \tag{30}$$

The solution using am(n) is the solution to the same minimization problem using a more elegant form.

Table 3: Forward and backward equations

Linear estimation (general)	Forward linear prediction of order m	Backward linear prediction of order m
$\sum_{i=1}^{n} \lambda^{n-i}\mathbf{u}(i)e^{*}(i) = \mathbf{0}$	$\sum_{i=1}^{n} \lambda^{n-i}\mathbf{u}_m(i-1)f_m^{*}(i) = \mathbf{0}$	$\sum_{i=1}^{n} \lambda^{n-i}\mathbf{u}_m(i)b_m^{*}(i) = \mathbf{0}$

At this point, we use equation 21 in equation 30 and next equation 23 and the condition of equation 29 to get the recursion equation,

$$F_m(n) = \lambda F_m(n-1) + \eta_m(n) f^*_m(n) \tag{31}$$

In this equation the product at the end is a real value.

V Adaptive Backward Linear Prediction [17]

Consider the backward linear predictor of order m. This is in Figure 12.5(a) for operation at time n. The tap weight vector is optimized using least squares sense until time n. Let [15].

$$b_m(i) = u(i-m) - \mathbf{w}^H_{eb,m}(n)\mathbf{u}_m(i)$$

$$i = 1, 2, \ldots, n, \tag{32}$$

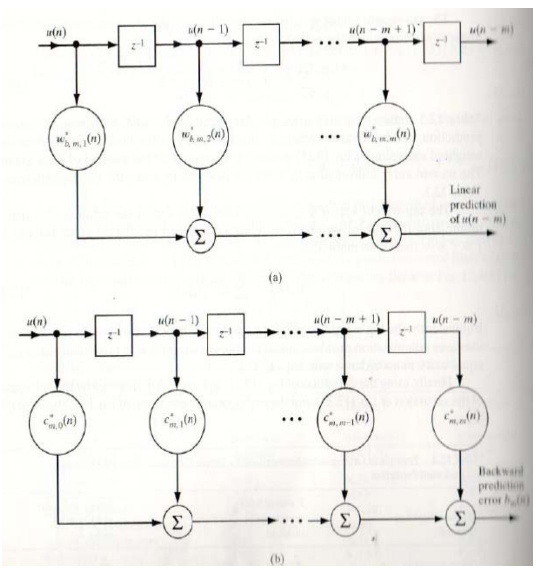

Figure 5: Backward prediction

This is the backward prediction error for the input vector um(i). We have

$$\mathbf{u(i)} = [u(i), u(i-1), \ldots\ldots, u(i-m+1)]^T$$

And

$$\mathbf{w}_{eb,m}(n) = [w_{eb,m,1}(n), w_{eb,m,2}(n), \ldots\ldots, w_{eb,m,m}(n)]^T$$

bm(i) is the backward a posteriori prediction error. It is dependent on the current value of the vector wb,m(n). we may define the backward a priori prediction error as

$$\beta_m(i) = u(i-m) - \mathbf{w}^H_{eb,m}(n-1)\mathbf{u}_m(i)$$

$$I = 1, 2, \ldots\ldots, n, \tag{33}$$

The computation is based on past weight vector wb,m(n).

To do recursion for adaptive backward linear prediction, we modify the RLS algorithm. The following is the recursion for updating the tap weight vector.

$$\mathbf{w}_{eb,m}(n) = \mathbf{w}_{eb,m}(n-1) + \mathbf{k}_m(n)\ \beta^*_m(n) \tag{34}$$

In equation 34 we have the backward priori prediction error and we have

$$\mathbf{k}_m(n) = \boldsymbol{\phi}^{-1}_m\mathbf{u}_m(n) \tag{35}$$

The matrix we have in equation 35 is the inverse of the correlation matrix

$$\boldsymbol{\phi}_m = \sum_{i=1}^{n} \lambda^{n-1}\mathbf{u}_m(i)\mathbf{u}_m^H(i) \tag{36}$$

We may analyze this problem as a backward prediction error filter problem. In this case, the tap weight vector is cm(n) which we can find from figure 12(b)as

$$\mathbf{c}_m(n) = \begin{bmatrix} -\boldsymbol{w} \\ 1 \end{bmatrix} \tag{37}$$

In this vector cm,m(n) is one and the input vector u m+1(i) of size m+1. In this case, the backward a posteriori prediction error and the backward a priori prediction error can be found as

$$b_m(i) = \mathbf{c}_m^H(n)\mathbf{u}_{m+1}(i)$$

$$i = 1, 2, \ldots, n, \tag{38}$$

$$\beta_m(i) = \mathbf{c}_m^H(n-1)\mathbf{u}_{m+1}(i)$$

$$i = 1, 2, \ldots\ldots, n, \tag{39}$$

The input vector is

$$\mathbf{u}_{m+1}(\mathbf{i}) = \begin{bmatrix} \boldsymbol{u(i)} \\ u(i-m) \end{bmatrix}$$

The tap weight vector is orthogonal to the backward linear prediction error. This mean

$$\sum_{i=1}^{n} \lambda^{n-1}\mathbf{u}_m(i)b_m^*(i) = \mathbf{0} \tag{40}$$

The tap weights vector wb,m(n) may also beseen as minimizing the sum

$$B_m(n) = \sum_{i=1}^{n} \lambda^{n-i}|b_m(i)| \tag{41}$$

for $1 < i < n$

Also, we can find cm(n) as a solution to the same minimization problem.

Using equation 32 in equation 41then equation34 and the orthogonality condition of equation 40 we get the recursion.

$$B_m(n) = \lambda B_m(n-1) + \beta_m(n)\ b_m^*(n) \tag{42}$$

To end this discussion, it is important to note in the case of backward prediction the input vector um+1(n) is partitioned with the desired response u(n-m) as the last entry. As in the case of forward prediction, the input vector um+1(n) is partitioned with u(n) as the first entry.

VI. Conversion Factor [18]

First, we defined the vector k as

$$\mathbf{k}_m(\mathbf{n}) = \boldsymbol{\phi}^{-1}_m(\mathbf{n})\ \mathbf{u}_m(\mathbf{n})$$

km(n) is the tap weight vector of the filter that operates on the data u(1), u(2)……..u(n) to produce the special response

$$d(i) = \begin{cases} 1 & i = n \\ 0 & i = 1, 2, \ldots\ldots\ldots, n-1 \end{cases} \qquad (43)$$

d(i) is an n by 1 vector, and the name of it is the first coordinate vector. This vector has the property that its dot product with any time-dependent vector is the last element of that vector.

First, we have to say that things are normalized. Second, we define the estimation error as

$$\begin{aligned} \gamma_m(n) &= 1 - \mathbf{k^H_m(n)}\ \mathbf{u_m(n)} \\ &= 1 - \mathbf{u^H_m(n)}\ \boldsymbol{\phi}_\mathbf{m}^{\mathbf{-1}}\mathbf{(n)}\ \mathbf{u_m(n)} \end{aligned} \qquad (44)$$

Were the estimation error is the output of the filter with tap weights km(n) and input um(n) as in figure 6. We can see from the equation 44 that the estimation error is real moreover it is between zero and one.

$$0 < \gamma_m(n) < 1 \qquad (45)$$

Know it is time to simplify things

$$\gamma_m(n) = 1 / [1 + \lambda^{-1}\mathbf{u_m^H(n)}\boldsymbol{\phi}_\mathbf{m}^{\mathbf{-1}}\mathbf{(n-1)u_m(n)}] \qquad (46)$$

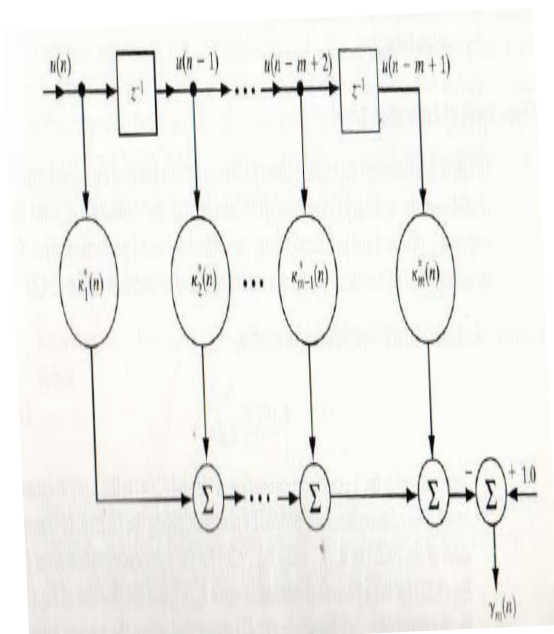

Figure 6: Conversion factor

Lambda between zero and one so the estimation error is bounded as in equation 45.

It is good to see that the estimation error is the output of the filter of figure 6 of the tap weight vector km(n).

VII. Some Useful Interpretation of the Estimation Error [14]

Depending on the way it is used the estimation error can have three different interpretations

1. The estimation error can be seen as the likelihood variable (Lee 1981). This is due to the statistical formulation of the tap input function in terms of its log-likelihood function. We say that the input has joint Gaussian distribution.
2. The estimation error can be seen as the angle variable (Lee 1981). This can be seen from equation 44. We may say

$$\gamma_m^{1/2}(n) = \cos\varphi_m(m)$$

Were phi is the angle of plane rotation.

3. The estimation error can be seen as the conversion factor (Carayannis 1983). It can be used to find an a posteriori estimation error from the a priori estimation error.

It is due to the third interpretation we use the term conversion factor.

VIII. Three Kinds of Estimation Error [14]

In linear least square estimation theory, we have three kinds of estimation error. The ordinary estimation error, the forward prediction error, and the backward prediction error. This means we have three interpretation as a conversion factor.

1. The recursive least squares estimation

Where we have the estimation error is equal to the posteriori error divided by the a priori estimation error. This can be seen from equation 44.

2. For adaptive forward linear prediction

$$\gamma_m(n-1) = f_m(n)/\eta_m(n) \qquad (48)$$

This can be seen by post-multiplying the Hermitian transposed sides of equation 23 by um(n-1) and then using equations 21 and 22 and 24 and 44.

3. For adaptive backward linear prediction

$$\gamma_m(n) = b_m(n)/\beta_m(n) \qquad (49)$$

As in 2 if we multiply equation 34 by um(n) and use equations 32 and 33 and 35 and 44 we can find 49. The estimation error can be seen as the multiplicative correction.

As we see the estimation error is the common factor (either regular or delayed) in the conversion from a priori to a posteriori estimation error. This is in ordinary estimation or forward prediction or backward prediction. We can use this conversion factor to find em(n) or fm(n) or bm(n) at time n before the tap weight has been computed (Carayannis 1983).

Global Journal of Computer Science and Technology (G) Volume XVII Issue III Version I Year 2017

IX. Least Square Lattice Predictor [13]

Using the time shifting property of the input data we write the partitioned vector.

$$\mathbf{u_{m+1}(n)}= \begin{matrix} \boldsymbol{u(n)} \\ u(n-m) \end{matrix}$$

We see that the input vector um(n) for the backward linear predictor of order m-1 and the input vector u(m+1)(n) for the backward linear predictor of order m have the same m-1 input entries. Let us move know to the partitioned vector.

$$\mathbf{u_{m+1}\ (n)}= \begin{matrix} u(n) \\ \boldsymbol{u(n-1)} \end{matrix}$$

The input vector um(n-1) for the forward linear predictor of order m-1 and the input vector u m+1 (n) for the forward linear predictor of order m have the same last m-1 entries. The question is can we carry over the information from stage m-1 to stage m.

The answer to this question is yes. And it employs modular structure known as lattice predictor.

To find this important filtering structure, we use the principle of orthogonality, and with the umbrella of Kalman filter theory, we find the least squares lattice predictor.

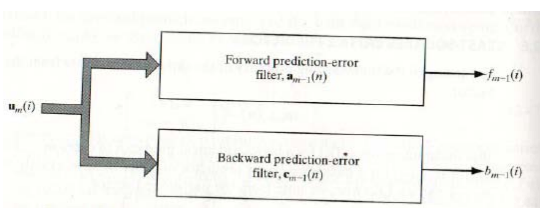

Figure 7: Block diagram

Let us begin with figure 7. The input is um(n). The upper part is a forward prediction error filter with tap weight vector a (m-1)(n) and output f (m-1)(i). The lower part is a backward prediction error filter with tap weight vector c (m-1)(n) and output b (m-1) (i). The problem we want to solve may be stated as.

Given the forward prediction error f (m-1)(i) and the backward prediction error b (m-1)(i) find their order update value f m (i) and b m (i) efficiently.

We mean by efficient manner is to use the information in f(m-1)(i) and b(m-1)(i) plus the input data is enlarged by the past sample u(i-m).

The past sample u(i-m) needed to compute fm(i) can be found from b (m-1)(i-1). Thus treating this as input to the one tap least square filter and f (m-1)(i) as the desired response and f m (i) as a result from least square approximation we can write

$$f_m(i) = f_{m-1}(i) + k^*_{f,m}(n)\, b_{m-1}(i-1)$$

$$i= 1, 2, \ldots\ldots, n, \qquad (50)$$

This is Figure 8

To find the coefficient of this filter we use the principal of orthogonality. According to this principal, the error produced by this filter f m (i) is orthogonal to the input b (m-1) (i).

$$\sum_{i=1}^{n} \lambda^{n-1} b_{m-1}(i-1) f^*_m(i) = 0 \qquad (51)$$

Substituting equation 50 into equation 51 and solving for the coefficient.

$$k_{f,m}(n) = \sum_{i=1}^{n} \lambda^{n-1} b_{m-1}(i-1) f^*_{m=1}(i)$$

$$/ \left[\sum_{i=1}^{n} \lambda^{n-1} |b_{m-1}(i-1)|^2\right] \qquad (52)$$

It is clear that

$$B_{m-1}(n-1)= \sum_{i=1}^{n} \lambda^{n-1} |b_{m-1}(i-1)|^2 \qquad (53)$$

Where in the last line we used the fact that

$$b_{m-1}(0) = 0 \quad \text{for all m} > 1$$

In equation 52 we have introduced the notation of exponentially weighted cross-correlation between forward and backward prediction error.

$$\Delta_{m-1}(n) = \sum_{i=1}^{n} \lambda^{n-1} b_{m-1}(i-1) f^*_{m-1}(i) \qquad (54)$$

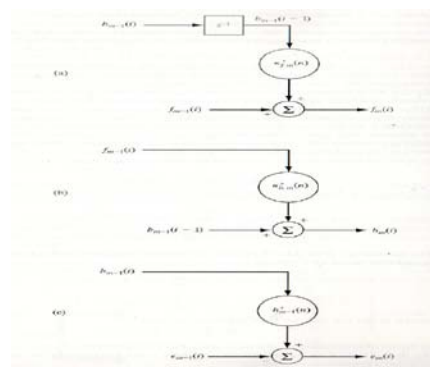

Figure 8: Recursion

Using equation 53 and equation 54 in equation 52 we see that the coefficient is

$$k_{f,m}(n) = \Delta_{m-1}(n) \ / \ B_{m-1}(n-1) \qquad (55)$$

We use the same method to find the order update for backward prediction error b m (i). The input is f(m-1)(i). The filter is figure 8 (b). It is clear that

$$b_m(i) = b_{m-1}(i-1) + k^*_{b,m}(n)\, f_{m-1}(i)$$

$$i= 1, 2, \ldots\ldots, n, \qquad (56)$$

Know it is time to determine the coefficient and to do this we use the orthogonality principal. The error b m(i) has to be orthogonal to the input f (m-1) (i). Thus we write

$$\sum_{i=1}^{n} \lambda^{n-1} f_{m-1}(i) b^*_m(i) = 0 \qquad (57)$$

Substituting equation 56 into equation 57 and solving for the coefficient.

$$k_{b,m}(n) = \sum_{i=1}^{n} \lambda^{n-1} f_{m-1}(i) b^*_{m-1}(i-1)$$

$$/ \left[\sum_{i=1}^{n} \lambda^{n-1} |f_{m-1}(i)|^2\right] \qquad (58)$$

Let us put

$$F_{m-1}(n) = \sum_{i=1}^{n} \lambda^{n-1} |f_{m-1}(i)|^2 \quad (59)$$

This mean equation 58 can be written as

$$k_{b,m}(n) = \Delta^{*}_{m-1}(n) \; / \; F_{m-1}(n) \quad (60)$$

Equation 50 and 56 are the basic to lattice predictor. For physical interpretation we define

$$\mathbf{f}_m(n) = [f_m(1), f_m(2), \ldots\ldots., f_m(n)]^T$$

$$\mathbf{b}_m(n) = [b_m(1), b_m(2), \ldots\ldots., b_m(n)]^T$$

$$\mathbf{b}_m(n-1) = [0, b_m(1), b_m(2), \ldots\ldots., b_m(n-1)]^T$$

Based on equation 50 and 56 we may make the following statements using the terminology of projection theory.

1. The result of projecting the vector b (m-1) (n-1) onto f (m-1) (n) is represented by the vector f m(n) and the forward reflection coefficient is the parameter needed to make this projection.
2. The result of projecting the vector f (m-1)(n) onto b (m-1)(n-1) is represented by the vector b(m)(n) . The back word reflection coefficient is the parameter needed to make this second projection.

So we have the pair of interrelated order update recursions.

$$f_m(n) = f_{m-1}(n) + k^{*}_{f,m}(n)\, b_{m-1}(n-1) \quad (61)$$

And

$$b_m(n) = b_{m-1}(n-1) + k^{*}_{f,m}(n)\, f_{m-1}(n) \quad (62)$$

m is the order of the filter and n is the time index. The initial condition is

$$f_0(n) = b_0(n) = u(n) \quad (63)$$

Where u(n) is the input at time n. And m is the prediction order from zero up to M. We have M stages least-squares lattice predictor in figure 9. An important feature is the lattice structure which implies linear complexity with the order.

X. Least Squares Lattice Version [13]

The forward prediction error and backward prediction error are determined by equations 27 and 38 as

$$f_m(n) = \mathbf{a_m^H(n) u_{m+1}(n)}$$

And

$$b_m(n) = \mathbf{c^H{}_m(n)} \; \mathbf{u_{m+1}(n)}$$

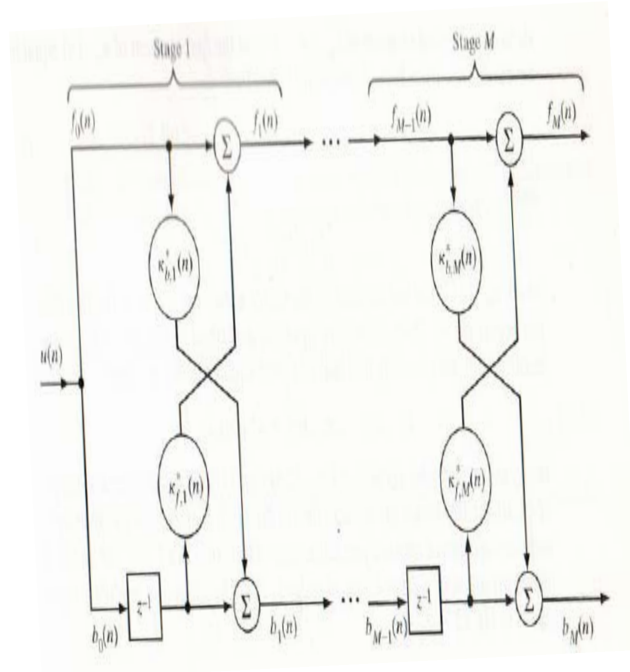

Figure 9: Lattice predictor

In the two equations, a m (n) and c m (n) are the tap weight vectors of the filters to calculate the backward and forward prediction error. The forward prediction error f (m-1)(n) and the backward prediction error b (m-1)(n) are defined as

$$f_m(n) = \mathbf{a_{m-1}{}^H(n) u_m(n)}$$

$$= \begin{bmatrix} \mathbf{a(m-1)(n)} \\ 0 \end{bmatrix}^H \begin{bmatrix} \mathbf{u(m)(n)} \\ u(n-m) \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{a(m-1)(n)} \\ 0 \end{bmatrix}^H \mathbf{u_{m+1}(n)}$$

$$b_{m-1}(n-1) = \mathbf{c_{m-1}{}^H(n-1) u_m(n-1)}$$

$$= \begin{bmatrix} 0 \\ \boldsymbol{c(m-1)(n-1)} \end{bmatrix}^H \begin{bmatrix} u(n) \\ \boldsymbol{u(m)(n-1)} \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ \boldsymbol{c(m-1)(n-1)} \end{bmatrix}^H \mathbf{u_{m+1}(n)}$$

The four prediction errors just defined have the same input u(m+1)(n). substituting in 61 and 62 and comparing terms we get

$$\mathbf{a_m(n)} = \begin{bmatrix} \mathbf{a(m-1)(n)} \\ 0 \end{bmatrix} + k_{f,m}(n) \begin{bmatrix} 0 \\ \boldsymbol{c(m-1)(n-1)} \end{bmatrix} \quad (64)$$

And

$$\mathbf{c_m(n)} = \begin{bmatrix} 0 \\ \boldsymbol{c(m-1)(n)} \end{bmatrix} + k_{f,m}(n) \begin{bmatrix} \boldsymbol{a(m-1)(n)} \\ 0 \end{bmatrix} \quad (65)$$

Equation 64 and equation 65 might be viewed as the least squares version of the Levinson Durbin recursion. Keeping in mind that the last element c (m-1)(n-1) and the first element a (m-1)(n) is equal to one. We see from 64 and 65 that

$$k_{f,m}(n) = a_{m,m}(n) \quad (66)$$

And

$$k_{b,m}(n) = c_{m,0}(n) \quad (67)$$

Where a m,m(n) is the last element of the vector am(n) and c m,0(n) is the first element of the vector cm(n). we generally find.

$$k_{f,m}(n) = k_{b,m}^{*}(n)$$

The order update equations 64 and 65 show a very good property of the lattice predictor of order M. we can say such a predictor have a chain of forward prediction error filters of order 1,2,………,M and a chain of backward prediction error filters of order 1,2,………,M all in one modular structure shown in figure 9.

XI. Time Update Recursion [17]

From equation 55 and 60 we find that the reflection coefficients (backward and forward)are uniquely determined by three quantities. Equation 31 and 32 provide the time update for two of them. We still have to find the time update equation for the third quantity (exponential cross-correlation).

To proceed, we recall the two equations with (m-1) in place of m.

$$f_{m-1}(i) = u(i) - \mathbf{w}^{*}_{f,m-1}(n)\ \mathbf{u}_{m-1}(\mathbf{i-1})$$

$$i = 1, 2, \ldots\ldots\ldots,n,$$

And

$$\mathbf{w}_{ef,m-1}(\mathbf{n})=\mathbf{w}_{ef,m-1}(\mathbf{n-1}) + \mathbf{k}_{m-1}(\mathbf{n-1})\ \eta_{m-1}^{*}(n)$$

Substituting in equation 54 we get

$$\Delta_{m-1}(n) =\sum_{i=1}^{n} \lambda^{n-1}[u(i)-\mathbf{w}_{ef,m-1}{}^{H}(\mathbf{n-1})\ \mathbf{u}_{m-1}(\mathbf{i-1})]^{*}b_{m-1}(i-1)$$

$$-\eta_{m-1}(n)\mathbf{k}^{T}{}_{m-1}(\mathbf{n-1})\sum_{i=1}^{n} \lambda^{n-1}b_{m-1}(i-1)\mathbf{u}^{*}{}_{m-1}(\mathbf{i-1})$$

This equation simplifies as follows,

First, the second term in the equation is zero using the principal of orthogonalization which states.

$$\sum_{i=1}^{n} \lambda^{n-1}\mathbf{u}_{m-1}(i-1)b^{*}{}_{m}(i) =\mathbf{0}$$

Second, the first term inside the brackets we have the a priori forward prediction error.

$$\eta_{m}(i)=u(i)-\mathbf{w}_{ef,\,m}(\mathbf{n-1})\ \mathbf{u}_{m}(\mathbf{i-1})$$

$$I = 1, 2, \ldots\ldots.,n.$$

This mean delta is

$$\Delta_{m-1}(n) =\sum_{i=1}^{n} \lambda^{n-1}b_{m-1}(i-1)\eta^{*}{}_{m-1}(i) \quad (68)$$

We can write this summation as

$$\Delta_{m-1}(n) =\sum_{i=1}^{n-1} \lambda^{n-1}b_{m-1}(i-1)\eta^{*}{}_{m-1}(i)$$

$$+ \ b_{m-1}(i-1)\eta^{*}{}_{m-1}(i)$$

$$\Delta_{m-1}(n) =\lambda\sum_{i=1}^{n-1} \lambda^{n-1-I}\ b_{m-1}(i-1)\eta^{*}{}_{m-1}(i)$$

$$+ \ b_{m-1}(i-1)\eta^{*}{}_{m-1}(i)$$

We know that the first term is simply delta (m-1)(n-1) so we write.

$$\Delta_{m}(n) = \lambda\ \Delta_{m}(n-1) + \eta^{*}{}_{m-1}(n)\ \ b_{m}^{*}(n-1) \quad (69)$$

Which is the desired equation. This is similar to equation 31 and 42 in that of these three updates the correction term has the product of posteriori and a priori prediction errors.

XII. Exact Decoupling Property of the Least Squares Lattice Predictor [18]

An important property of this predictor is that the backward prediction errors at different stages are uncorrelated. This is plus that they are orthogonal. Keep in mind that the input u(n) might be a correlated sequence. This means we are transforming a correlated sequence to uncorrelated one.

$$[u(n), u(n-1), \ldots\ldots\ldots, u(n-m)]$$

$$\leftrightarrow [b_0(n), b_1(n), \ldots\ldots., b_m(n)] \quad (70)$$

The transformation here is reciprocal which mean that this filter keeps the information content of the input data.

The tap weight vector of the filter is cm(n)

$$\mathbf{c}_{m}(\mathbf{n}) = [c_{m,m}(n), c_{m,m-1}(n), \ldots\ldots., 1]$$

We want to find the backward a posteriori prediction error bm(i) using the input u(m+1)(i).

$$\mathbf{u}_{m+1}(\mathbf{i}) =[u(i), u(i-1), \ldots\ldots\ldots, u(i-m)]$$

$$i> m$$

We can express bm(i) as

$$b_{m}(i) = \mathbf{c}^{H}{}_{m}(\mathbf{m})\mathbf{u}_{m-1}(\mathbf{i})$$

$$= \sum_{k=0}^{m} c^{*}{}_{m,k}(n)u(i-m+k)$$

$$m < i < n$$

$$m = 1, 2, \ldots. \quad (71)$$

Let

$$\mathbf{b}_{m+1}(\mathbf{i}) = [b_0(n), b_1(n), \ldots\ldots., b_m(i)]_T$$

$$m < i < n$$

$$m = 1, 2, \ldots.$$

Be (m+1) by 1 backward a posteriori prediction error vector. Substituting equation 71 into this vector we have the transformation [19]

$$\mathbf{b}_{m+1}(\mathbf{i}) = \mathbf{L}_{m}(\mathbf{n})\ \mathbf{u}_{m+1}(\mathbf{i}) \quad (72)$$

Where the m+1 by m+1 transformation matrix

$$\mathbf{L}_{m}(\mathbf{n}) = \begin{matrix} 1 & 0 & 0 \\ c(1,1)(n) & 1 & 0 \\ c(m,m)(n) & c(m,m-1)(n) & 1 \end{matrix} \quad (73)$$

This is a lower triangular matrix. It is an m by m matrix and note the following.

1. A non zero element of row l in the matrix Lm(n) is the tap weight of the backward prediction filter of order (l-1).
2. The diagonal elements of Lm(n) are equal to unity. This is because the last tap weight of this filter equals unity.
3. The determinant of the matrix Lm(n) is one for all m.

This mean the inverse matrix exist. This means that the reciprocal nature of equation 70 is confirmed.

The correlation between the backward prediction errors of orders k and m is zero.

Using the principal of orthogonality, it is clear that the error bm(i) is perpendicular to the input uk(i) and this means that the correlation is zero for m not equal k. This means that bm(n) and bk(n) are uncorrelated in the time-averaged sense.

This property makes this system an ideal device for exact least squares joint process estimation. We might use the sequence of bm(n) in figure 9 to perform the least squares estimation of the desired response as in figure 10. We may write

$$e_m(n) = e_{m-1}(n) - h^*_{m-1}(n) \; b_{m-1}(n)$$

$$m = 1, 2, \ldots\ldots, M+1 \qquad (75)$$

The initial condition of the joint process estimation is

$$e_o(n) = d(n) \qquad (76)$$

The parameter h(m-1)(n) are called joint process estimation or regression coefficients. Thus the estimation of the desired response d(n) may go as a stage by stage basis, jointly with the linear prediction process.

Equation 75 is shown in figure 8(c). We use i in the figure to be consistent with 8(a) and 8(b). the input is b(m-1)(i) and the desired response is e(m-1)(i).[18].

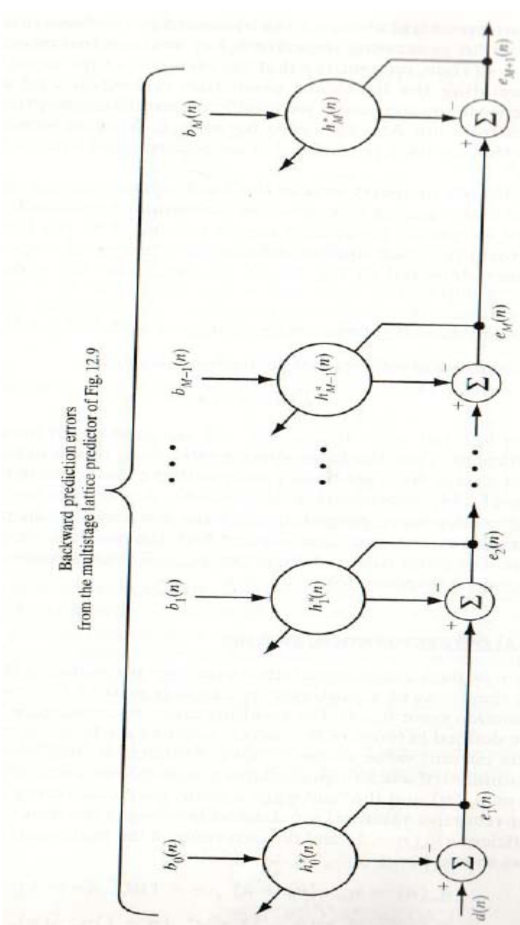

Figure 10: Correction

It is a desire to put the lattice problem not in term of the posteriori or a priori errors. This introduces the notation of angel.

XIII. Simulation Results

In this part, we will use mat lab. The desired response is an output of a Wiener filter of the first order and coefficient a=.3. The input is random signal. This input is given to the Wiener filter and the lattice predictor also first order. We feed the desired signal d(n) to the lattice predictor. The block diagram of the system is figure 11. As we can see from the simulation results, the coefficient h1 will pick up the value of a=.3 of the Wiener filter (figure 12).

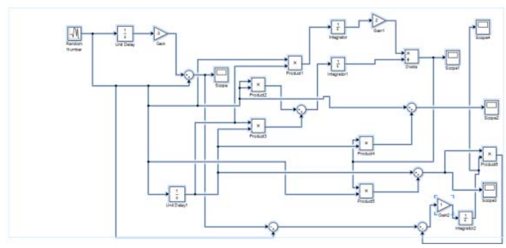

Figure 11: Mat Lab simulation

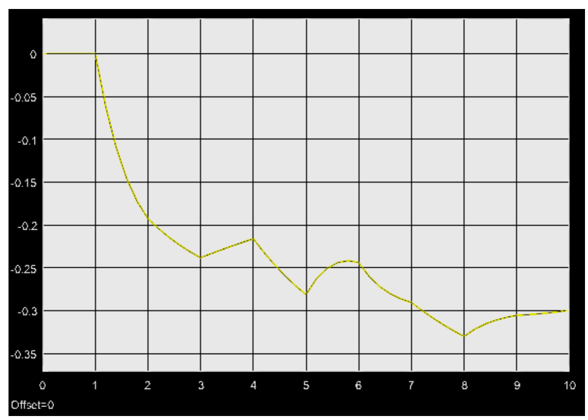

Figure 12: Simulation results

References Références Referencias

1. Papoulis, Probability, Random Variables, and Stochastic Process, 2002.
2. J. G. Proakis, Digital Communications, 2001
3. R. J. Schilling, Engineering Analysis ,1988
4. H. L. Van Trees, Detection, Estimation, and Modulation Theory, 1968.
5. J. G, Proakis, Inroduction to Digital Signal Processing, 1988.
6. Chen, Linear System Theory and Design, 1984.
7. S. Haykin, Communication System,1983.
8. T. H. Glisson, Introduction to System Analysis, 1985.
9. Martin Schetzen, Airborne Doppler Radar, 2006.
10. Martin Schetzen, The Volterra & Wiener Theories of Nonlinear Systems, 2006.

11. Martin Schetzen, Discrete System using Matlab, 2004.
12. Arvin Grabel, Microelectronics, 1987.
13. Ziadsobih, Time and Space, (International Journal of Engineering),Volume (7) : Issue (3) : 2013.
14. Ziadsobih, Construction of the Sampled Signal up to any Frequency While Keeping the Sampling Rate Fixed. (Signal Processing International Journal), Volume (7): Issue (2): 2013.
15. Ziadsobih, Up/Down Converter Linear Model With Feed Forward And Feedback Stability Analysis, (International Journal of Engineering), Volume (8): Issue (1): 2014.
16. Ziadsobih, Generation of any Pdf from A Set of Equally Likely Random Variables, Global Journal of Computer Science and Technology, Volume (14): Issue (2) : Version 1.0 : Year 2014.
17. Ziadsobih, Adaptive Filters, Global Journal of Researches in Engineering (F) Volume (14): Issue (7): Version (1): Year 2014
18. Ziadsobih, an Adaptive Filter to Pick Up A Wiener Filter from the Error with and without Noise, Global Journal of Researches in Engineering (F) Volume (15): Issue (2): Version (1): Year 2015.
19. Simon Haykin, Adaptive Filter Theory, 2001.