National Symposium on Hydrology
Dec.16-18,1987, N.I.H., Roorkee

WEIGHTED ORDINARY LEAST SQUARE ALGORITHM FOR BATCH PROCESSING

RAKESH KHOSA[1]

ABSTRACT

A deconvolution algorithm for a linear system with finite memory is developed using exponential weighting. The algorithm is developed with the idea of using it in batch mode. The solution derived is similar in nature to the ordinary least squares solution of the deconvolution problem for linear systems. The limitation of the existing procedures whereby analysis is based on a policy of equal weighting for all measured data as the process evolves is highlighted. It is suggested that this algorithm may be more suitable in total response modelling type of analysis than in a unit hydrograph type of study where events are treated in isolation.

Introduction:

The ordinary least squares algorithm is based on a policy of equal weighting for all measured data as the process evolves. The reason for using equal weighting was that the parameters were essentially constant throughout the period of estimation so that the most recent data was as good as older data for providing information about the unknown parameter values. However, when this algorithm is applied to a situation where the parameters to be estimated are time varying, the estimates can easily become erratic and do not bear a close resemblence to the true time variation of the parameter values(1).

---

[1]Pool Officer, Water Resources Management Centre, Deptt. of Civil Engineering, Regional Engineering College, Hazratbal, Srinagar, Kashmir (J&K).

Derived below is a least squares algorithm in which an exponential weighting scheme is used to place heavier emphasis on the more recent data. This algorithm is designed to be used in situations where observed rainfall and runoff data must be used in batch mode as distinct from "on line" mode. In "on line" system identification problems where the parameter values are estimated recursively, an initial estimate of these parameter values is required which in then progressively updated. It is suggested, therefore, that the algorithm derived below may be used in tandem with an "on line" identification algorithm where the initial parameter estimates required by the latter may be provided by the former.

WEIGHTED ORDINARY LEAST SQUARES ALGORITHM

Consider a linear time invariant system having a finite memory m with concurrent input and output series of length n and denoted by x_i and y_i ($i = 1, 2, \ldots n$), respectively. It may also be assumed that in general x_1 and y_i ($i=1,2,\ldots n$) may not represent an independent and isolated event. Representing the series x_i and y_i schematically as shown below, it is understood that a consequence of the system having a finite memory m will be

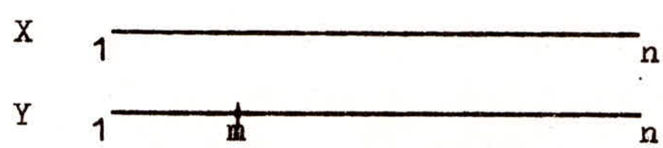

Fig.1

the existence of a linear time invariant relationship between y_i ($i =1,2, \ldots m-1$) and those elements of the input series x_i which correspond to measurement times i prior to the commencement of measurement i.e. $i < 1$. It therefore becomes imperative for the purpose of analysis, to neglect the first ($m-1$) values of the output series y_i($i=1,2,\ldots n$) and seek a relationship between x_i($i=1,2, \ldots n$) and y_i($i=m,m+1,\ldots n$)[2].

Let the estimate of the pulse response function $\underset{\sim}{h}$ be denoted as

$$\hat{\underset{\sim}{h}} = [\hat{h}_1 \ \hat{h}_2 \ \hat{h}_3 \ \dots \ \hat{h}_m]^T \qquad \dots (1)$$

(Note $\underset{\sim}{h}$ is the true pulse response function to be estimated and '$\sim$' denotes that the variable is an array and not a scalar)

An estimate of the output y_n is then given by

$$\hat{y}_n = x_n \hat{h}_1 + X_{n-1} \hat{h}_2 + X_{n-2} \hat{h}_3 + \dots + X_{n-m+1} \hat{h}_m$$

$$= [X_n \ \ X_{n-1} \ \ X_{n-2} \ \dots \ X_{n-m+1}] \begin{bmatrix} \hat{h}_1 \\ \hat{h}_2 \\ \hat{h}_3 \\ \vdots \\ \hat{h}_m \end{bmatrix}$$

Denote the row vector above as

$$\underset{\sim}{f}_n = (X_n \ \ X_{n-1} \ \ X_{n-2} \ \dots \ X_{n-m+1}) \qquad \dots (2)$$

[in general $\underset{\sim}{f}_{i+1} = (X_{i+1} \ X_i \ X_{i-1} \ \dots \ X_{i-m+2})$]

The error in the estimation of y_n is then given by

$$e_n = (y_n - \underset{\sim}{f}_n \hat{\underset{\sim}{h}})$$

In ordinary least squares with equal weighting, an attempt is made to minimise the objective function

$$J_n(\hat{\underset{\sim}{h}}) = \sum_{i=m}^{n} (y_i - \underset{\sim}{f}_i \hat{\underset{\sim}{h}})^2$$

Consider the error function[3)]

$$J_n(\hat{\underset{\sim}{h}}) = \sum_{i=m}^{n} (y_i - \underset{\sim}{f}_i \hat{\underset{\sim}{h}})^2 \lambda^{n-i} \qquad \dots (3)$$

$$0 < \lambda \leq 1$$

From equation (3) the following points may be noted:

a) If $\lambda = 1$, equation (3) reduces to ordinary least squares with equal weighting.

b) Smaller the value of λ , the heavier are the weights assigned to the more recent data.

Equation (3) may be rewritten as

$$J_n(\hat{h}) = \sum_{i=m}^{n} (y_i^2 - 2y_i \underset{\sim}{f_i} \underset{\sim}{\hat{h}} + (\underset{\sim}{f_i}\underset{\sim}{\hat{h}})^2) \lambda^{n-i}$$

$$= \sum_{i=m}^{n} y_i^2 \lambda^{n-i} - 2 \sum_{i=m}^{n} y_i \underset{\sim}{f_i} \underset{\sim}{\hat{h}} \lambda^{n-i} + \sum_{i=m}^{n} (\underset{\sim}{f_i} \underset{\sim}{\hat{h}})^2 \lambda^{n-i}$$

For a minimum $\frac{dJ_n}{d\underset{\sim}{\hat{h}}}(\hat{h}) = 0$

$$\frac{dJ_n(\hat{h})}{d\underset{\sim}{\hat{h}}} = \frac{d}{d\underset{\sim}{\hat{h}}}(-2 \sum_{i=m}^{n} y_i \underset{\sim}{f_i} \underset{\sim}{\hat{h}} \lambda^{n-i}) + \frac{d}{d\underset{\sim}{\hat{h}}}(\sum_{i=m}^{n} (\underset{\sim}{f_i} \underset{\sim}{\hat{h}})^2 \lambda^{n-i} \quad \ldots(4)$$

Considering the first term on the R.H.S of equation (4) we have

$$\sum_{i=m}^{n} y_i \underset{\sim}{f_i} \underset{\sim}{\hat{h}} \lambda^{n-i} = y_m \underset{\sim}{f_m} \underset{\sim}{\hat{h}} \lambda^{n-m} + y_{m+1} \underset{\sim}{f_{m+1}} \underset{\sim}{\hat{h}} \lambda^{n-(m+1)} + \ldots + y_n \underset{\sim}{f_n} \underset{\sim}{\hat{h}} \lambda^{n-n}$$

(Note λ^{n-i} and y_i's are scalars)

Also $$f_m \underset{\sim}{\hat{h}} = (x_m \; x_{m-1} \; x_{m-2} \cdots x_1) \begin{bmatrix} \hat{h}_1 \\ \hat{h}_2 \\ \hat{h}_3 \\ \vdots \\ \hat{h}_m \end{bmatrix}$$

$$= x_m\hat{h}_1 + x_{m-1}\hat{h}_2 + x_{m-2}\hat{h}_3 + \ldots + x_1\hat{h}_m$$

or we may write

$$y_m \underset{\sim}{f_m} \underset{\sim}{\hat{h}} \lambda^{n-m} = y_n \lambda^{n-m} (x_m\hat{h}_1 + x_{m-1}\hat{h}_2 + x_{m-2}\hat{h}_3 + \ldots + x_1\hat{h}_m)$$

and $$\frac{d}{d\hat{h}_1}(y_m \underset{\sim}{f_m} \underset{\sim}{\hat{h}} \lambda^{n-m}) = \lambda^{n-m} y_m x_m$$

$$\frac{d}{d\hat{h}_2}(y_m \underset{\sim}{f_m} \underset{\sim}{\hat{h}} \lambda^{n-m}) = \lambda^{n-m} y_m x_{m-1}$$

$$\vdots$$

$$\frac{d}{d\hat{h}_m}(y_m \underset{\sim}{f_m} \underset{\sim}{\hat{h}} \lambda^{n-m}) = \lambda^{n-m} y_m x_1$$

Therefore

$$\frac{d}{d\hat{\underset{\sim}{h}}}(y_m \underset{\sim}{f}_m \hat{\underset{\sim}{h}} \lambda^{n-m}) = \lambda^{n-m} \; y_m \; \underset{\sim}{f}_m^T$$

Similarly

$$\frac{d}{d\hat{\underset{\sim}{h}}}(y_{m+1} \underset{\sim}{f}_{m+1} \hat{\underset{\sim}{h}} \lambda^{n-(m+1)}) = \lambda^{n-(m+1)} \; y_{m+1} \; \underset{\sim}{f}_{m+1}^T$$

$$\vdots$$

$$\frac{d}{d\hat{\underset{\sim}{h}}}(y_n \underset{\sim}{f}_n \hat{\underset{\sim}{h}} \lambda^{n-n}) = \lambda^{n-n} \; y_n \; \underset{\sim}{f}_n^T$$

Assembling these terms, the first term on the R.H.S of equation (4) becomes

$$\frac{d}{d\hat{\underset{\sim}{h}}}\left(-2\sum_{i=m}^{n} y_i \underset{\sim}{f}_i \hat{\underset{\sim}{h}} \lambda^{n-i}\right) = -2\left[\lambda^{n-m} y_m \underset{\sim}{f}_m^T + \lambda^{n-(m+1)} y_{m+1} \underset{\sim}{f}_{m+1}^T + \dots + \lambda^{n-n} y_n f_n^T\right]$$

or $$\frac{d}{d\hat{\underset{\sim}{h}}}\left(-2\sum_{i=m}^{n} y_i \underset{\sim}{f}_i \hat{\underset{\sim}{h}} \lambda^{n-i}\right) = -2\sum_{i=m}^{n} y_i \underset{\sim}{f}_i^T \lambda^{n-i} \qquad \text{-----} \quad (5)$$

Similarly for the second term on the R.H.S of equation(4) we may write

$$\sum_{i=m}^{n} (\underset{\sim}{f}_i \hat{\underset{\sim}{h}})^2 \lambda^{n-i} = (\underset{\sim}{f}_m \hat{\underset{\sim}{h}})^2 \lambda^{n-m} + (\underset{\sim}{f}_{m+1} \hat{\underset{\sim}{h}})^2 \lambda^{n-(m+1)} + \dots -- + (\underset{\sim}{f}_n \hat{\underset{\sim}{h}})^2 \lambda^{n-n}$$

It may be noted that $\underset{\sim}{f}_i \hat{\underset{\sim}{h}}$ $(i=m, m+1, \dots n)$ is a scalar.

Therefore

$$\frac{d}{d\hat{\underset{\sim}{h}}}\left(\sum_{i=m}^{n} (\underset{\sim}{f}_i \hat{\underset{\sim}{h}})^2 \lambda^{n-i}\right) = \frac{d}{d\hat{\underset{\sim}{h}}}(\underset{\sim}{f}_m \hat{\underset{\sim}{h}})^2 \lambda^{n-m} + \frac{d}{d\hat{\underset{\sim}{h}}}(\underset{\sim}{f}_{m+1} \hat{\underset{\sim}{h}})^2 \lambda^{n-(m+1)} + \dots + \frac{d}{d\hat{\underset{\sim}{h}}}(\underset{\sim}{f}_n \hat{\underset{\sim}{h}})^2 \lambda^{n-n}$$

also

$$\frac{d}{d\hat{\underset{\sim}{h}}}(\underset{\sim}{f}_m \hat{\underset{\sim}{h}})^2 \lambda^{n-m} = \frac{d}{d\hat{\underset{\sim}{h}}}(\underset{\sim}{f}_m \hat{\underset{\sim}{h}})(\underset{\sim}{f}_m \hat{\underset{\sim}{h}}) \lambda^{n-m}$$

assume $\underset{\sim}{f}_m = (x_m \; x_{m-1})$ and $\hat{\underset{\sim}{h}} = \begin{bmatrix}\hat{h}_1 \\ \hat{h}_2\end{bmatrix}$

$$(\underset{\sim}{f}_m \hat{\underset{\sim}{h}})^2 = \left[(x_m \; x_{m-1})\begin{pmatrix}\hat{h}_1 \\ \hat{h}_2\end{pmatrix}\right]^2 = (x_m \hat{h}_1 + x_{m-1}\hat{h}_2)^2$$

$$= x_m^2 \hat{h}_1^2 + 2 X_m \hat{h}_1 x_{m-1} \hat{h}_2 + X_{m-1}^2 \hat{h}_2^2$$

and

$$\frac{d}{d\hat{h}_1}(\underset{\sim}{f}_m \hat{\underset{\sim}{h}})^2 = 2 X_m^2 \hat{h}_1 + 2 X_m X_{m-1} \hat{h}_2$$

$$\frac{d}{d\hat{h}_2}(\underset{\sim}{f}_m\underset{\sim}{\hat{h}})^2 = 2\, X_{m-1}^2\, \hat{h}_2 + 2X_m\, X_{m-1}h_1,$$

or

$$\frac{d}{d\underset{\sim}{\hat{h}}}(\underset{\sim}{f}_m\underset{\sim}{\hat{h}})^2 = 2 \left[\begin{pmatrix} x_m \\ x_{m-1} \end{pmatrix} (\hat{h}_1\ \hat{h}_2) \begin{pmatrix} x_m \\ x_{m-1} \end{pmatrix} \right]$$

$$= 2\, \underset{\sim}{f}_m^T\, \underset{\sim}{\hat{h}}^T\, \underset{\sim}{f}_m^{\ T}$$

Similarly if

$$\underset{\sim}{f}_m = [X_m\ X_{m-1}\ X_{m-2}] \text{ and } \underset{\sim}{\tilde{h}} = \begin{bmatrix} \hat{h}_1 \\ \hat{h}_2 \\ \hat{h}_3 \end{bmatrix}$$

$$(\underset{\sim}{f}_m\underset{\sim}{\hat{h}})^2 = (X_m\hat{h}_1 + X_{m-1}\hat{h}_2 + X_{m-2}\hat{h}_3)^2$$

$$= X_m^2\,\hat{h}_1^2 + X_{m-1}^2\,\hat{h}_2^2 + 2\,X_m\,\hat{h}_1\,X_{m-1}\hat{h}_2$$

$$+ 2\,X_m\,\hat{h}_1\,X_{m-2}\hat{h}_3$$

$$+ X_{m-2}^2\,\hat{h}_3^2 + 2\,X_{m-1}\,\hat{h}_2\,X_{m-2}\,\hat{h}_3$$

and

$$\frac{d}{d\hat{h}_1}(\underset{\sim}{f}_m\underset{\sim}{h})^2 = 2\,X_m^2\,\hat{h}_1 + 2\,X_m\,X_{m-1}\,\hat{h}_2 + 2\,X_m\,X_{m-2}\,\hat{h}_3$$

$$\frac{d}{d\hat{h}_2}(\underset{\sim}{f}_m\underset{\sim}{\hat{h}})^2 = 2\,X_{m-1}^2\,\hat{h}_2 + 2\,X_m\,X_{m-1}\hat{h}_1 + 2\,X_{m-1}X_{m-2}\hat{h}_3$$

$$\frac{d}{d\hat{h}_3}(\underset{\sim}{f}_m\underset{\sim}{\hat{h}})^2 = 2\,X_{m-2}^2\hat{h}_3 + 2X_m\,X_{m-2}\hat{h}_1 + 2X_{m-1}X_{m-2}\hat{h}_2$$

or $$\frac{d}{d\underset{\sim}{\hat{h}}}(\underset{\sim}{f}_m\underset{\sim}{\hat{h}})^2 = 2 \left[\begin{pmatrix} X_m \\ {}_{m-1} \\ X_{m-2} \end{pmatrix} (\hat{h}_1\ \hat{h}_2\ \hat{h}_3) \begin{pmatrix} X_m \\ X_{m-1} \\ X_{m-2} \end{pmatrix} \right]$$

$$= 2\ \underset{\sim}{f}_m^T\ \underset{\sim}{\hat{h}}^T\ \underset{\sim}{f}_m^T$$

By induction, therefore, for $\underset{\sim}{f}_m = (X_m\ X_{m-1}\ X_{m-2}\ \cdots\ X_1)$

$$\frac{d}{d\underset{\sim}{\hat{h}}}(\underset{\sim}{f}_m\underset{\sim}{\hat{h}})^2\lambda^{\,n-m} = 2\lambda^{n-m}\ \underset{\sim}{f}_m^T\,\underset{\sim}{\hat{h}}^T\,\underset{\sim}{f}_m^T$$

$$\frac{d}{d\underset{\sim}{\hat{h}}}(\underset{\sim}{f}_{m+1}\underset{\sim}{\hat{h}})^2\lambda^{\,n-(m+1)} = 2\lambda^{n-(m+1)}\ f_{m+1}^T\ \ h^T\, f_{m+1}^T$$

$$\frac{d}{d\underset{\sim}{\hat{h}}}(\underset{\sim}{f}_n\underset{\sim}{\hat{h}})^2\lambda^{n-n} = 2\lambda^{n-n}\,\underset{\sim}{f}_n^T\,\underset{\sim}{\hat{h}}^T\,\underset{\sim}{f}_n^T$$

Therefore the second term on the R.H.S of equation (4) becomes

$$\frac{d}{d\underset{\sim}{\hat{h}}}\left(\sum_{i=m}^{n}(\underset{\sim}{f}_i\underset{\sim}{\hat{h}})^2\lambda^{n-i}\right) = 2\sum_{i=m}^{n}\underset{\sim}{f}_i^T\;\underset{\sim}{\hat{h}}^T\;\underset{\sim}{f}_i^T\,\lambda^{n-i} \quad \ldots (6)$$

Substituting (5) and (6) in equation(4), we have

$$\frac{d}{d\underset{\sim}{\hat{h}}}J_n(\underset{\sim}{\hat{h}}) = -2\sum_{i=m}^{n}y_i\,\underset{\sim}{f}_i^T\,\lambda^{n-i} + 2\sum_{i=m}^{n}\underset{\sim}{f}_i^T\,\underset{\sim}{\hat{h}}^T\,\underset{\sim}{f}_i^T\lambda^{n-i}$$

For a minimum $\dfrac{dJ_n(\underset{\sim}{\hat{h}})}{d\underset{\sim}{\hat{h}}} = 0$

$$\text{or}\quad \sum_{i=m}^{n}\underset{\sim}{f}_i^T\;\underset{\sim}{\hat{h}}^T\;\underset{\sim}{f}_i^T\lambda^{n-i} = \sum_{i=m}^{n}y_i\;\underset{\sim}{f}_i^T\;\lambda^{n-i} \quad \ldots (7)$$

Note that
$$\sum_{i=m}^{n}\underset{\sim}{f}_i^T\;\underset{\sim}{\hat{h}}^T\;\underset{\sim}{f}_i^T\lambda^{n-i} = \underset{\sim}{f}_m^T\,\underset{\sim}{\hat{h}}^T\underset{\sim}{f}_m^T\lambda^{n-m} + \underset{\sim}{f}_{m+1}^T\,\underset{\sim}{\hat{h}}^T\underset{\sim}{f}_{m+1}^T\lambda^{n-(m+1)} + \ldots + \underset{\sim}{f}_n^T\,\underset{\sim}{\hat{h}}^T\,\underset{\sim}{f}_n^T\lambda^{n-n}$$

$$= \begin{bmatrix}\underset{\sim}{f}_m^T & \underset{\sim}{f}_{m+1}^T & \underset{\sim}{f}_{m+2}^T & \cdots & \underset{\sim}{f}_n^T\end{bmatrix}\begin{bmatrix}\underset{\sim}{\hat{h}}^T\,\underset{\sim}{f}_m^T\,\lambda^{n-m}\\ \underset{\sim}{\hat{h}}^T\,\underset{\sim}{f}_{m+1}^T\lambda^{n-(m+1)}\\ \vdots\\ \underset{\sim}{\hat{h}}^T\,\underset{\sim}{f}_n^T\lambda^{n-n}\end{bmatrix}$$

Since $\underset{\sim}{\hat{h}}^T\,\underset{\sim}{f}_m^T$ is a Scalar $\quad\underset{\sim}{\hat{h}}^T\,\underset{\sim}{f}_m^T = \underset{\sim}{f}_m\,\underset{\sim}{\hat{h}}$

$$\text{or}\quad \sum_{i=m}^{n}\underset{\sim}{f}_i^T\,\underset{\sim}{\hat{h}}^T\,\underset{\sim}{f}_i^T\lambda^{n-i} = \begin{bmatrix}\underset{\sim}{f}_m^T & \underset{\sim}{f}_{m+1}^T & \underset{\sim}{f}_{m+2}^T & \cdots & \underset{\sim}{f}_n^T\end{bmatrix}\begin{bmatrix}\lambda^{n-m}\,\underset{\sim}{f}_m\,\underset{\sim}{\hat{h}}\\ \lambda^{n-(m+1)}\underset{\sim}{f}_{m+1}\underset{\sim}{\hat{h}}\\ \lambda^{n-(m+2)}\underset{\sim}{f}_{m+2}\underset{\sim}{\hat{h}}\\ \vdots\\ \lambda^{n-n}\,\underset{\sim}{f}_n\,\underset{\sim}{\hat{h}}\end{bmatrix} \quad \ldots (8)$$

Now

$$\left[\underset{\sim}{f}_m^T \; \underset{\sim}{f}_{m+1}^T \cdots \underset{\sim}{f}_n^T\right] = \begin{bmatrix} X_m & X_{m+1} & X_{m+2} & \cdots & X_n \\ X_{m-1} & X_m & X_{m+1} & \cdots & X_{n-1} \\ X_{m-2} & X_{m-1} & X_m & \cdots & X_{n-2} \\ \vdots & \vdots & \vdots & & \vdots \\ X_1 & X_2 & X_3 & \cdots & X_{n-m+1} \end{bmatrix} \quad \ldots (9)$$

order : $[\, m \times (n-m+1) \,]$

Denote the matrix in (9) above as $[\underset{\sim}{X}_1]$ and $\lambda^{n-(m+i)}$ as A_i

The column vector in (8) may be written as

$$\begin{bmatrix} \lambda^{n-m} \underset{\sim}{f}_m \underset{\sim}{\hat{h}} \\ \lambda^{n-(m+1)} \underset{\sim}{f}_{m+1} \underset{\sim}{\hat{h}} \\ \vdots \\ \lambda^{n-n} \underset{\sim}{f}_n \underset{\sim}{\hat{h}} \end{bmatrix} = \begin{bmatrix} A_0 & (X_m \; X_{m-1} \; X_{m-2} \cdots X_1) \\ A_1 & (X_{m+1} \; X_m \; X_{m-1} \cdots X_2) \\ A_2 & (X_{m+2} \; X_{m+1} \; X_m \cdots X_3) \\ \vdots & \\ A_{n-m} & (X_n \; X_{n-1} \; X_{n-2} \cdots X_{n-m+1}) \end{bmatrix} \begin{bmatrix} \hat{h}_1 \\ \hat{h}_2 \\ \hat{h}_3 \\ \vdots \\ \hat{h}_m \end{bmatrix}$$

Denote the matrix above as $[\underset{\sim}{X}_2]$ i.e.,

$$\begin{bmatrix} A_0 X_m & A_0 X_{m-1} & A_0 X_{m-2} & \cdots & A_0 X_1 \\ A_1 X_{m+1} & A_1 X_m & A_1 X_{m-1} & \cdots & A_1 X_2 \\ A_2 X_{m+2} & A_2 X_{m+1} & A_2 X_m & \cdots & A_2 X_3 \\ A_{n-m} X_n & A_{n-m} X_{n-1} & A_{n-m} X_{n-2} & \cdots & A_{n-m} X_{n-m+1} \end{bmatrix} = [\underset{\sim}{X}_2] \quad \ldots (10)$$

order : $[\,(n-m+1) \times m\,]$

Equation(8) therefore, becomes

$$\sum_{i=m}^{n} \underset{\sim}{f}_i^T \hat{\underset{\sim}{h}}^T \underset{\sim}{f}_i^T \lambda^{n-i} = [\underset{\sim}{X}_1]^T [\underset{\sim}{X}_2] [\hat{\underset{\sim}{h}}] \qquad \ldots(11)$$

Consider, now, the R.H.S. of equation (7)

$$\sum_{i=m}^{n} y_i \underset{\sim}{f}_i^T \lambda^{n-i} = y_m \underset{\sim}{f}_m^T \lambda^{n-m} + y_{m+1} \underset{\sim}{f}_{m+1}^T \lambda^{n-(m+1)} + \ldots + y_n \underset{\sim}{f}_n^T \lambda^{n-n}$$

Since y_i and λ_i are scalars, we may write

$$\sum_{i=m}^{n} y_i \underset{\sim}{f}_i^T \lambda^{n-i} = \lambda^{n-m} \underset{\sim}{f}_m^T y_m + \lambda^{n-(m+1)} \underset{\sim}{f}_{m+1}^T y_{m+1} + \lambda^{n-(m+2)} \underset{\sim}{f}_{m+2}^T y_{m+2} + \ldots + \lambda^{n-n} \underset{\sim}{f}_n^T y_n$$

$$= [A_0 \underset{\sim}{f}_m^T \; A_1 \underset{\sim}{f}_{m+1}^T \; A_2 \underset{\sim}{f}_{m+2}^T \ldots A_{n-m} \underset{\sim}{f}_n^T] \begin{bmatrix} y_m \\ y_{m+1} \\ y_{m+2} \\ \vdots \\ y_n \end{bmatrix}$$

Note that $[A_0 \underset{\sim}{f}_m^T \; A_1 \underset{\sim}{f}_{m+1}^T \; A_2 \underset{\sim}{f}_{m+2}^T \ldots A_{n-m} \underset{\sim}{f}_n^T] = [\underset{\sim}{X}_2]^T$

where $[\underset{\sim}{X}_2]$ is as given by equation (10).

Denoting the vector $\begin{bmatrix} y_m \\ y_{m+1} \\ y_{m+2} \\ \vdots \\ y_n \end{bmatrix}$ as $[\underset{\sim}{Y}]$

We may, therefore, write

$$\sum_{i=m}^{n} y_i \underset{\sim}{f}_i^T \lambda^{n-i} = [\underset{\sim}{X}_2]^T [\underset{\sim}{Y}] \qquad \ldots(12)$$

Substituting (11) and (12) in equation (7), we get

$$\left[\underset{\sim}{X_1}\right]^T \left[\underset{\sim}{X_2}\right] \left[\underset{\sim}{\hat{h}}\right] = \left[\underset{\sim}{X_2}\right]^T \left[\underset{\sim}{Y}\right] \qquad \ldots (13)$$

Let us understand the nature of the matirx

$$\left[\underset{\sim}{X_1}\right]^T \left[\underset{\sim}{X_2}\right]$$

$$\left[\underset{\sim}{X_1}\right]^T \left[\underset{\sim}{X_2}\right] = \begin{pmatrix} X_m & X_{m+1} & X_{m+2} \cdots & X_n \\ X_{m-1} & X_m & X_{m+1} \cdots & X_{n-1} \\ X_{m-2} & X_{m-1} & X_m \cdots & X_{n-2} \\ \vdots & \vdots & \vdots & \vdots \\ X_1 & X_2 & X_3 \cdots & X_{n-m+1} \end{pmatrix} \begin{pmatrix} A_0X_m & A_0X_{m-1} & \cdots A_0X_1 \\ A_1X_{m+1} & A_1X_m & \cdots A_1X_2 \\ A_2X_{m+2} & A_2X_{m+1} & . A_2X_3 \\ \vdots & \vdots & \vdots \\ A_{n-m}X_n & A_{n-m}X_{n-1} & \cdots A_{n-m}X_{n-m+1} \end{pmatrix}$$

$$\begin{bmatrix} \sum\limits_{j=m}^{n} A_{j-m}X_j^2 & \sum\limits_{j=m}^{n} A_{j-m}X_jX_{j-1} & \sum\limits_{j=m}^{n} A_{j-m}X_jX_{j-2} & \cdots \sum\limits_{j=m}^{n} A_{j-m}X_jX_{j-m+1} \\ \sum\limits_{j=m}^{n} A_{j-m}X_jX_{j-1} & \sum\limits_{j=m-1}^{n-1} A_{j-m+1}X_j^2 & \sum\limits_{j=m-1}^{n-1} A_{j-m+1}X_jX_{j-1} & \cdots \sum\limits_{j=m-1}^{n-1} A_{j-m+1}X_jX_{j-m+2} \\ \sum\limits_{j=m}^{n} A_{j-m}X_jX_{j-2} & \sum\limits_{j=m-1}^{n-1} A_{j-m+1}X_jX_{j-1} & \sum\limits_{j=m-2}^{n-2} A_{j-m+2}X_j^2 & \cdots \sum\limits_{j=m-2}^{n-2} A_{j-m+2}X_jX_{j-m+3} \\ \vdots & & & \\ \sum\limits_{j=m}^{n} A_{j-m}X_jX_{j-m+1} & \sum\limits_{j=m-1}^{n} A_{j-m+1}X_jX_{j-m+2} & \sum\limits_{j=m-2}^{n-2} A_{j-m+2}X_jX_{j-m+3} & \cdots \sum\limits_{j=1}^{n-m+1} A_{j-1}X_j^2 \end{bmatrix}$$

...(14)

The matrix given in equation (14) is a square matrix of order (mxm). It can be seen to be a symmetric matrix and the general form of the elements of this matrix is as given below

$$\left\{ [\underset{\sim}{X_1}]^T [\underset{\sim}{X_2}] \right\}_{i,j} = \sum_{k=1}^{n-m+1} A_{k-1} \; X_{(m+k-i)} \; X_{(m+k-j)} \quad \ldots (15)$$

$$i=1,2,\ldots m$$
$$j=1,2,\ldots m$$

where i and j denote the row and column number respectively.

Equation (13) may also be written in the more familiar form as

$$[\underset{\sim}{\hat{h}}] = \left\{ [\underset{\sim}{X_1}]^T [\underset{\sim}{X_2}] \right\}^{-1} [\underset{\sim}{X_2}]^T [\underset{\sim}{Y}] \ldots (16)$$

It can similarly be shown that the vector $[\underset{\sim}{X_2}]^T [\underset{\sim}{Y}]$ has the following structure.

$$[\underset{\sim}{X_2}]^T [\underset{\sim}{Y}] = \begin{bmatrix} \sum_{j=m}^{n} A_{j-m} \; X_j y_j \\ \sum_{j=m}^{n} A_{j-m} \; X_{j-1} y_j \\ \sum_{j=m}^{n} A_{j-m} \; X_{j-2} y_j \\ \vdots \\ \sum_{j=m}^{n} A_{j-m} \; X_{j-m+1} \; y_j \end{bmatrix}$$

and is of order (m x 1)

The general form of the elements of this vector are as given below

$$\left\{ [\underset{\sim}{X_2}]^T [\underset{\sim}{Y}] \right\}_i = \sum_{k=1}^{n-m+1} A_{k-1} \; X_{(m+k-i)} \; Y_{(m+k-1)} \quad \ldots (17)$$

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

Procedure:

To implement the algorithm as given by equations (15), (16) and (17), one may proceed as follows:

1) Choice of λ. This will depend on the problem at hand and may have to be chosen arbitrarily with the refinement done by trial and error process on the basis of its performance in the calibration period.

2) Using equation (15) the elements of the matrix $[\underset{\sim}{X}_1]^T [\underset{\sim}{X}_2]$ may be estimated from the data and the matrix itself built up.

3) The vector $[\underset{\sim}{X}_2]^T [\underset{\sim}{Y}]$ may be built up from the elements estimated as per equation (17).

4) Use equation (16) to get the estimate $[\hat{\underset{\sim}{h}}]$ of the system pulse response function $\hat{\underset{\sim}{h}}$.

Conclusion:

The algorithm developed above uses exponential weighting with maximum weight given to more recently measured input and output. These weights decrease exponentially for data measured in the past. The assumption of a time invariant system in the context of total response modelling may not be justified for hydrologic systems. This view is borne out by the fact that the requirements and rigours of an ever expanding population, various developmental activities and a consequent degration of the environment manifests itself in a varying system response. It is felt, therefore, that the algorithm developed above may be expected, in general, to give better and more representative results than the ordinary least squares algorithm. It must be noted, however, that where a unit hydrograph type of study is to be carried out in which attention is focussed on individual isolated events and considering the fact that duration of such events is much smaller (of the order of just a few days), weighted least squares algorithm as developed above may not be needed.

Acknowledgement:

The author is deeply indebted to all the brilliant creators of the past and the present day who volunteered to carry the world on their shoulders.

References:

(1) Hsia, T.C. (1977). System Identification. 29-30 pp. Lexington Books.

(2) Khosa, R. (1986). Applications of Toeplitz structure in the Least Squares Solution of the Deconvolution problem for linear systems. Unpublished M.Sc dissertation. Deptt. of Engg.Hydrology,University College, Galway, Ireland.

(3) Bruen, M. (1984-85). Numerical Methods in Hydrology. Unpublished Lecture notes. Deptt. of Engg. Hydrology, University College, Galway, Ireland.