Journal of Signal and Information Processing, **2011, 2, 287-291**
doi:10.4236/jsip.2011.24041 Published Online November 2011 (http://www.SciRP.org/journal/jsip)

Least Squares Matrix Algorithm for State-Space Modelling of Dynamic Systems

Juuso T. Olkkonen[1], **Hannu Olkkonen**[2]

[1]VTT Technical Research Centre of Finland, Espoo, Finland; [2]Department of Applied Physics, University of Eastern Finland, Kuopio, Finland.
Email: Juuso.Olkkonen@vtt.fi

Received October 4th, 2011; revised November 1st, 2011; accepted November 10th, 2011.

ABSTRACT

This work presents a novel least squares matrix algorithm (*LSM*) *for the analysis of rapidly changing systems using state-space modelling. The LSM algorithm is based on the Hankel structured data matrix representation. The state transition matrix is updated without the use of any forgetting function. This yields a robust estimation of model parameters in the presence of noise. The computational complexity of the LSM algorithm is comparable to the speed of the conventional recursive least squares* (*RLS*) *algorithm. The knowledge of the state transition matrix enables feasible numerical operators such as interpolation*, *fractional differentiation and integration. The usefulness of the LSM algorithm was proved in the analysis of the neuroelectric signal waveforms.*

Keywords: *State-Space Modelling*, *Dynamic System Analysis*, *EEG*

1. Introduction

Estimation of the state of the dynamic systems has been a research object for many years since the innovation of the Kalman filter (KF) [1-4]. The time-varying autoregressive models are useful in analysis of relatively slowly changing dynamic systems. The adaptive least mean squares (LMS) algorithm has been extensively applied in the analysis of various biomedical and industrial systems [5,6]. A disadvantage of the LMS algorithm is the poor adaptation in systems with abrupt changes. The more fastly adapting recursive least squares (RLS) algorithm [7], Kohonen neural network [8], extended Kalman filter (EKF) [9] and many other approaches have recently introduced for robust and accurate space-state modelling of the highly varying dynamic systems. The computational power of the most of the algorithms is based on the recursive updating of the model parameters and matrices. Usually this is solved by using a forgetting function, which gives the higher weight to the most recent data values.

In this work we describe the least squares matrix (LSM) algorithm, where the state-space model is based on the Hankel structured data matrix formulation. In the updating algorithm, no forgetting function is used.

2. Theoretical Considerations

2.1. Dynamic State-Space Model

We consider the dynamic state-space model

$$X_{n+1} = F_n X_n, \quad y_n = C X_n + w_n, \tag{1}$$

where the state vector $X_n \in \mathbb{R}^{N\times 1}$, the state transition matrix $F_n \in \mathbb{R}^{N\times N}$ and the vector $C = [1 0 0 \cdots 0] \in \mathbb{R}^{1\times N}$. The scalar $w_n \in \mathbb{R}^{1\times 1}$ is a random zero mean observation noise. The signal $y_n \in \mathbb{R}^{1\times 1}$ consists of measurements at time increments $t = nT \left(n = 0,1,2,\cdots\right)$, where T is the sampling period. Let us define the data vector $Y_n = \left[y_n \; y_{n-1} \cdots y_{n-N-1}\right]^T$ and the Hankel structured data matrix H_n as

$$H_n = \begin{bmatrix} y_n & y_{n-1} & \cdots & y_{n-M-1} \\ y_{n-1} & y_{n-2} & \cdots & y_{n-N-2} \\ \vdots & \vdots & \vdots & \vdots \\ y_{n-N-1} & y_{n-N-2} & \cdots & y_{n-N-M-2} \end{bmatrix} = \left[Y_n \quad Y_{n-1} \quad \cdots \quad Y_{n-M-1}\right]. \tag{2}$$

The subscript n in Y_n and H_n refers to the most recent data point y_n. The least squares estimate of the state transition matrix G_n comes from

$$\begin{aligned} H_{n+1} &= F_n H_n \;\Rightarrow \\ F_n &= H_{n+1} H_n^T (H_n H_n^T)^{-1} \\ &= H_{n+1} H_n^{\#} = R_n C_n^{-1}, \end{aligned} \tag{3}$$

where the pseudoinverse matrix $H_n^{\#} = H_n^T \left(H_n H_n^T\right)^{-1} \in \mathbb{R}^{N\times N}$, $R_n = H_{n+1} H_n^T \in \mathbb{R}^{N\times N}$ and $C_n = H_n H_n^T \in \mathbb{R}^{N\times N}$. The rank of the state transition matrix F_n defines the system order. In many applications the state transition matrix should be evaluated at T intervals. In complex dynamic systems the dimension of the state transition matrix is high and the computation of the pseudoinverse matrix $H_n^{\#}$ is time consuming. Instead, by partitioning the state transition matrix F_n into R_n and C_n^{-1} matrices we introduce a novel algorithm for updating the C_n^{-1} and R_n matrices and for the computation of the state transition matrix F_n.

2.2. Computation of the C_n^{-1} Matrix

Using the data vector representation (2) the C_n matrix can be written as

$$C_n = H_n H_n^T = Y_n Y_n^T + Y_{n-1}Y_{n-1}^T + \cdots + Y_{n-M-1}Y_{n-M-1}^T. \quad (4)$$

The uptake of the C_n matrix is obtained by adding a new term and subtracting the oldest term as

$$C_{n+1} = C_n + Y_{n+1}Y_{n+1}^T - Y_{n-M-1}Y_{n-M-1}^T. \quad (5)$$

A key idea in this work is that we write the last two terms as

$$Y_{n+1}Y_{n+1}^T - Y_{n-M-1}Y_{n-M-1}^T = \begin{bmatrix} Y_{n+1} & Y_{n-M-1} \end{bmatrix} \begin{bmatrix} Y_{n+1}^T \\ -Y_{n-M-1}^T \end{bmatrix} = U_n\, V_n, \quad (6)$$

where $U_n \in \mathbb{R}^{N\times 2}$ and $V_n \in \mathbb{R}^{2\times N}$. Now we obtain the updated matrix C_{n+1} as

$$C_{n+1} = C_n + U_n\, V_n. \quad (7)$$

The matrix inversion lemma yields the inverse matrix

$$\begin{aligned} C_{n+1}^{-1} &= (C_n + U_n V_n)^{-1} \\ &= C_n^{-1} - C_n^{-1} U_n (I + V_n C_n^{-1} U_n)^{-1} V_n C_n^{-1}, \end{aligned} \quad (8)$$

where the identity matrix $I \in \mathbb{R}^{2\times 2}$. By denoting $Z_n = \left(I + V_n C_n^{-1} U_n\right)^{-1} V_n C_n^{-1} \in \mathbb{R}^{2\times N}$ we have

$$C_{n+1}^{-1} = C_n^{-1} - C_n^{-1} U_n Z_n. \quad (9)$$

For fast computation of the Z_n matrix the product $R_n = V_n C_n^{-1} \in \mathbb{R}^{2\times N}$ is first computed and then the inverse matrix $\left(I + R_n U_n\right)^{-1} \in \mathbb{R}^{2\times 2}$. In Equation (9) the product $C_n^{-1} U_n \in \mathbb{R}^{N\times 2}$ is first computed.

2.3. Computation of the R_n Matrix

The matrix R_n can be written as

$$\begin{aligned} R_n &= H_{n+1} H_n^T \\ &= Y_{n+1}Y_n^T + Y_n Y_{n-1}^T + \cdots + Y_{n-M}Y_{n-M-1}^T. \end{aligned} \quad (10)$$

The matrix R_n can be updated as

$$\begin{aligned} R_{n+1} &= H_{n+2}H_{n+1}^T = R_n + Y_{n+2}Y_{n+1}^T - Y_{n-M}Y_{n-M-1}^T \\ &= R_n + \begin{bmatrix} Y_{n+2} & Y_{n-M} \end{bmatrix} \begin{bmatrix} Y_{n+1}^T \\ -Y_{n-M-1}^T \end{bmatrix} = R_n + U_{n+1} V_n, \end{aligned} \quad (11)$$

where the same notations as in Equation (6) have been used.

2.4. Computation of the F_n Matrix

The uptake of the state transition matrix $F_n = R_n\, C_n^{-1}$ comes from

$$\begin{aligned} F_{n+1} &= R_{n+1} C_{n+1}^{-1} \\ &= \left(R_n + U_{n+1} V_n\right)\left(C_n^{-1} - C_n^{-1} U_n Z_n\right) \\ &= F_n - F_n U_n Z_n + U_{n+1} V_n C_{n+1}^{-1}. \end{aligned} \quad (12)$$

For fast computation the products $F_n U_n \in \mathbb{R}^{N\times 2}$ and $V_n C_{n+1}^{-1} \in \mathbb{R}^{2\times N}$ are first computed. The uptake of the state transition matrix needs five matrix multiplications dimensioned as $\left(N\times N\right)\times\left(N\times 2\right)$ and four matrix multiplications dimensioned as $\left(N\times 2\right)\times\left(2\times N\right)$ or $\left(2\times N\right)\times\left(N\times 2\right)$. Thus the computational complexity of the algorithm is $5\,O\left(2N^2\right) + 4\,O\left(4N\right)$.

3. Applications of the LSM Algorithm

3.1. State-Space Filtering

The knowledge of the state transition matrix F_n enables the filtering of the measurement signal y_n based on the prediction

$$\hat{H}_{n+1} = F_n H_n \quad (13)$$

The state-space filtered signal $\hat{y}_n$ can be obtained as a mean of the antidiagonal elements. We may define the filtered data matrix as

$$\hat{H}_n = \begin{bmatrix} \hat{y}_n & \hat{y}_{n-1} & \cdots & \hat{y}_{n-M-1} \\ \hat{y}_{n-1} & \hat{y}_{n-2} & \cdots & \hat{y}_{n-M-2} \\ \vdots & \vdots & \vdots & \vdots \\ \hat{y}_{n-N-1} & \hat{y}_{n-N-2} & \cdots & \hat{y}_{n-N-M-2} \end{bmatrix}. \quad (14)$$

In the following we describe several matrix operators based on the state transition matrix. In all computations the filtered data matrix (14) is applied.

3.2. Numerical Signal Processing

The knowledge of the state transition matrix F_n enables the numerical signal processing of the state-space filtered signal. In the following we develop matrix operators based on the state transition matrix for numerical interpolation, differentiation and integration of the measurement signal.

The eigenvalue decomposition of the state transition matrix is $F_n = U_n D_n U_n^{-1}$, where $D_n \in \mathbb{R}^{N\times N} = \mathrm{diag}\left(\lambda_1\ \lambda_2 \cdots \lambda_N\right)$ and $U_n \in \mathbb{R}^{N\times N}$. Based on (14) we have a result

$$\begin{aligned} \hat{H}_{n+1} &= F_n \hat{H}_n = U_n D_n U_n^{-1} \hat{H}_n \Rightarrow \\ \hat{H}_{n+\Delta} &= U_n D_n^{\Delta} U_n^{-1} \hat{H}_{n+\Delta} = F_n^{\Delta} \hat{H}_n, \end{aligned} \quad (15)$$

where the time-shift $\Delta \in [0, T]$. Now we may define the interpolating operator $S_{n,\Delta} \in \mathbb{R}^{N\times N}$ as

$$\hat{H}_{n+\Delta} = S_{n,\Delta}\hat{H}_n \Rightarrow S_{n,\Delta} = F_n^{\Delta}. \quad (16)$$

Next, we may define the differentiation operator $D_n \in \mathbb{R}^{N\times N}$ as

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{H}_n = D_n\hat{H}_n \Rightarrow \hat{H}_{n+1} = e^{D_n}\hat{H}_n. \quad (17)$$

Due to Equation (15) we have

$$F_n = e^{D_n} \Rightarrow D_n = \log m(F_n), \quad (18)$$

where $\log m(\cdot)$ denotes matrix logarithm. Further, we may define an integral operator $I_n \in \mathbb{R}^{N\times N}$ as

$$\int \hat{H}_n \mathrm{d}t = I_n\hat{H}_n. \quad (19)$$

Since the differentiation and integral operator are inverse operators

$$I_n\hat{H}_n = D_n^{-1}\hat{H}_n = \left[\log m(F_n)\right]^{-1}\hat{H}_n \Rightarrow \\ I_n = \left[\log m(F_n)\right]^{-1} \quad (20)$$

The interpolating, differentiation and integral operators are commutative, *i.e.* $S_nD_n = D_nS_n$ and $S_nI_n = I_nS_n$. The computation of the second, third etc. derivatives and integrals of the signals are also possible using the matrix operators, e.g. the second derivative operator is obtained as $D_n^2 = \left[\log m(F_n)\right]^2$. Generally, the present method allows the fractional calculus, for example the computation of fractional derivative $D_n^{\alpha} = \left[\log m(F_n)\right]^{\alpha}$, where $\alpha \geq 0$. It should be pointed out that applied to the state-space filtered signals the numerical operators are analytic, *i.e.* they produce results with machine precision.

4. Experimental Results

The tracking performance of the LSM algorithm was tested using different kind of sinusoidal waveforms. In the absence of noise the outcome of the LMS algorithm followed the original signal with the machine precision. **Figure 1** shows the tracking performance of the LSM algorithm to a sinusoidal signal consisting of two frequency components ($\omega_1 = 0.1$ and $\omega_2 = 0.2$) in the presence of zero mean impulsive noise. The length of the data window was $M = 40$ and the system order $N = 4$. The mean error in estimation of the frequency components based on the eigenvalues of the state transition matrix was 0.14%. For comparison, the state transition matrix was computed from the pseudoinverse matrix (3) using the singular value decomposition (SVD) $H_n = U\Sigma V^T$, where U and V are unitary and Σ a diagonal matrix consisting of the singular values in descending order. The pseudoinverse matrix is then yielded as $H_n^{\#} = V\Sigma^{-1}U^T$, where the smallest singular values are eliminated [10,11]. An excellent match was found between the LSM and the SVD-based algorithms.

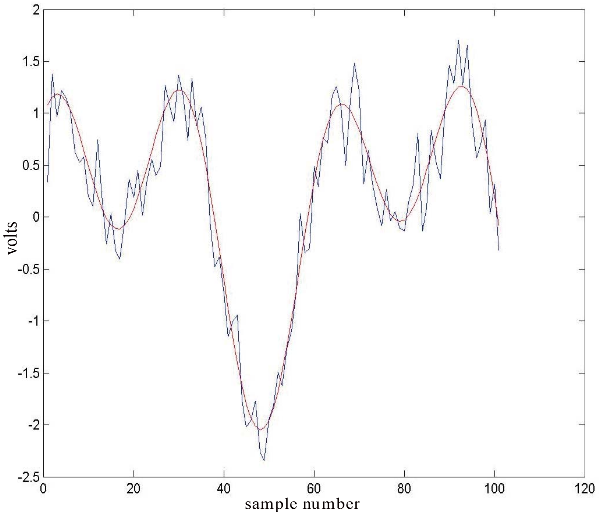

Figure 1. The tracking of the LSM algorithm to the noisy signal consisting of two sinusoidal components. The vertical scale is in volts and the horizontal scale denotes the sample number.

The validity of the LSM algorithm was warranted in the analysis of neuroelectric signal waveforms. The neuroelectric signals were recorded from two different locations of the brain in freely behaving Kuo-Wistar rats (bred in the National Animal Center of Kuopio, Finland): 1) from the frontal cortex with a stainless steel screw electrode (diameter 1 mm) driven into the scull bone, the electrode tip locating in the vicinity of the epipial neocortical surface, 2) from the hilar region of the dentate gyrus of the hippocampus with a permanently fixed depth surface wire electrode (diameter 125 µm, stainless steel, Nylon coated, tip cut square). In both recordings ground and reference points were situated on opposite sides of occipital skull. The neuroelectric signals were sampled at 300 Hz using a 14 bit analog-to-digital (ADC) converter. In front of the ADC any analog filter was not used.

In 16 consecutive EEG recordings the neuronal activities computed by the LSM and SVD methods cross-correlated highly significantly with each other. The cross-correlation coefficient varied between 0.999 - 0.9999. A typical neuroelectric signal recording from the hilar region and the outcome of the LSM algorithm is described in **Figure 2**. The waveform is mixed with varying degrees of irregular behavior derived from sources of nonsynchronously bursting neural activities. The eigenvalue decomposition of the state transition matrix yielded four clearly different frequency components with time-varying amplitudes.

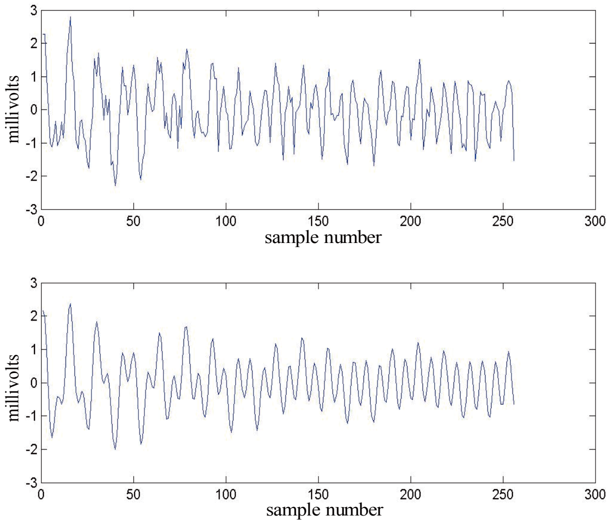

Figure 2. The original neuroelectric signal (top) and the outcome of the LMS algorithm (below). The x-axes denote the sample number. The y-axes denote the signal voltage in millivolts.

5. Conclusions

In this work we describe the least squares matrix (LSM) algorithm for the estimation of the dynamic state-space models. In conventional recursive least squares (RLS) algorithms uptake is based on the use of the forgetting factor, which weights the data vectors by an exponenttially descending function. In the present algorithm the rectangular weighting function is used, where the Hankel data matrix includes *M* vectors (4). The uptake of the data matrix consists of the addition of the most recent data vector Y_{n+1} and subtraction of the latest vector Y_{n-M-1}. This leads to a novel uptake mechanism (7) via U_n and V_n matrices, which are dimensioned as $U_n \in \mathbb{R}^{N\times 2}$ and $V_n \in \mathbb{R}^{2\times N}$. Due to the reduced dimensions of the U_n and V_n matrices, the computations needed in the uptake have computational complexity, which is of the same order than in the conventional RLS algorithm, which is usually referred as $O\left(N^2\right)$.

The RLS algorithm is known to be only marginally stable in the analysis of the fastly time varying systems. In the present algorithm the estimation of the state transition matrix is based on the LS solution of the Hankel structured data matrix, which consists of M data vectors. The method is inherently robust, since any adaptive filtering criteria are not used. The computation time does not depend on the number of data vectors *M*. The length of the data vectors *N* matches the system order. For noise free measurements *M* may be only slightly higher than *N*. In analysis of systems corrupted with noise the increase of *M* makes the algorithm more noise tolerant. On the other hand the overdeterministic solution masks the rapid changes in system parameters.

The good tracking performance of the LSM algorithm was warranted in the state-space modelling of the sinusoidal signals (**Figure 1**) and the neuroelectric signal waveforms (**Figure 2**). The outcome of the LSM algorithm correlated well with the results yielded by SVD method. The small differences are probably due to the fact that in the SVD method the smallest singular values must be eliminated before the computation of the pseudoinverse matrix. This reduces the system order in the SVD method. In the LSM algorithm the system order can be higher and the state-space modelling is more tolerant to variations in system parameters. The distinct difference between the present algorithm and the SVD based methods is that the present algorithm updates the state transition matrix F_n at every time interval, while the SVD based algorithms [10,11] compute the state transition matrix in data blocks. Our algorithm is more feasible in the analysis of the fastly changing dynamic systems and especially for real-time applications, where the eigenvalues of the state transition matrix give actual information on the system functioning.

The knowledge of the state transition matrix yields a plenty of numerical signal processing tools, such as interpolation (16), differentiation (18) and integration operators (20), which compete for example with the conventional B-spline signal processing algorithms [12-14].

6. Acknowledgements

We are indebted to the anonymous reviewer, whose comments improved the manuscript significantly.

REFERENCES

[1] F. Daum, “Nonlinear Filters: Beyond the Kalman Filter,” *IEEE A&E Systems Magazine*, Vol. 20, No. 8, 2005, pp. 57-69.

[2] A. Moghaddamjoo and R. L. Kirlin, “Robust Adaptive Kalman Filtering with Unknown Inputs,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 37, No. 8, 1989, pp. 1166-1175. doi:10.1109/29.31265

[3] J. L. Maryak, J. C. Spall and B. D. Heydon, “Use of the Kalman Filter for Interference in State-Space Models with Unknown Noise Distributions,” *IEEE Transactions on Automatic Control*, Vol. 49, No. 1, 2005, pp. 87-90.

[4] R. Diversi, R. Guidorzi and U. Soverini, “Kalman Filtering in Extended Noise Environments,” *IEEE Transactions on Automatic Control*, Vol. 50, No. 9, 2005, pp. 1396-1402. doi:10.1109/TAC.2005.854627

[5] S. Attallah, “The Wavelet Transform-Domain LMS Adaptive Filter with Partial Subband-Coefficient Updating,” *IEEE Transactions on Circuits and Systems II*: *Express Briefs*, Vol. 53, No. 1, 2006, pp. 8-12. doi:10.1109/TCSII.2005.855042

[6] H. Olkkonen, P. Pesola, A. Valjakka and L. Tuomisto,

"Gain Optimized Cosine Transform Domain LMS Algorithm for Adaptive Filtering of EEG," *Computers in Biology and Medicine*, Vol. 29, 1999, pp. 129-136. doi:10.1016/S0010-4825(98)00046-8

[7] E. Eweda, "Comparison of RLS, LMS, and Sign Algorithms for Tracking Randomly Time-Varying Channels," *IEEE Transactions on Signal Processing*, Vol. 43, No. 11, 1994, pp. 2937-2944. doi:10.1109/78.330354

[8] D. Niebur and A. J. Germond, "Power System Static Security Assessment Using the Kohonen Neural Network Classifier," *IEEE Transactions on Power Systems*, Vol. 7, No. 2, 1992, pp. 865-872. doi:10.1109/59.141797

[9] D.-J. Jwo and S.-H. Wang, "Adaptive Fuzzy Strong Tracking Extended Kalman Filtering for GPS Navigation," *IEEE Sensors Journal*, Vol. 7, No. 5, 2007, pp. 778-789. doi:10.1109/JSEN.2007.894148

[10] S. Park, T. K. Sarkar and Y. Hua, "A Singular Value Decomposition-Based Method for Solving a Deterministic Adaptive Problem," *Digital Signal Processing*, Vol. 9, No. 1, 1999, pp. 57-63. doi:10.1006/dspr.1998.0331

[11] T. J. Willink, "Efficient Adaptive SVD Algorithm for MIMO Applications," *IEEE Transactions on Signal Processing*, Vol. 56, No. 2, 2008, pp. 615-622. doi:10.1109/TSP.2007.907806

[12] M. Unser, A. Aldroubi and M. Eden, "B-Spline Signal Processing. I. Theory," *IEEE Transactions on Signal Processing*, Vol. 41, No. 2, 1993, pp. 821-833. doi:10.1109/78.193220

[13] M. Unser, A. Aldroubi and M. Eden, "B-Spline Signal Processing. II. Efficiency Design and Applications," *IEEE Transactions on Signal Processing*, Vol. 41, No. 2, 1993, pp. 834-848. doi:10.1109/78.193221

[14] J. T. Olkkonen and H. Olkkonen, "Fractional Time-Shift B-Spline Filter," *IEEE Signal Processing Letters*, Vol. 14, No. 10, 2007, pp. 688-691. doi:10.1109/LSP.2007.896402