A FAST INSTRUMENTAL VARIABLE AFFINE PROJECTION ALGORITHM

Karim Maouche and Dirk T.M. Slock

Institut Eurécom, Mobile Communication Department
2229, route des Crêtes, B.P. 193, 06904 Sophia Antipolis Cedex, France.
E-mail: maouche,slock@eurecom.fr

ABSTRACT

We derive a new adaptive filtering algorithm called the Instrumental Variable Affine Projection (IVAP) algorithm and give its fast version (FIVAP algorithm). The IVAP algorithm departs from the AP algorithm and uses an IV. The IV process is generated in a way such that the new algorithm combines between the AP and the Fast Newton Transversal Filter (FNTF) algorithms. Simulations show that the IVAP algorithm is more robust to noise than the AP algorithm. With the IV, the sample covariance matrix loses its Hermitian property and its displacement structure is different from the one of the AP algorithm. Consequently, the derivation of a fast version is done by deriving the IV Sliding Window Covariance Fast Transversal Filter (IV SWC FTF) algorithm. Using this and other ingredients, we derive the FIVAP algorithm whose computational complexity is nearly the same as the one of the FAP algorithm.

1. INTRODUCTION

The Fast Affine Projection algorithm outperforms the classical adaptive algorithms because of its convergence speed which approaches that of the Recursive Least Squares (RLS) algorithm and its computational complexity which is slightly greater than the one of the Least Mean Squares (LMS) algorithm. The AP algorithm is characterized by an updating-projection scheme of the adaptive filter on an L dimensional data-related subspace. This projection on a subspace whose dimension is in general very small compared to the filter length, gives the AP algorithm a tracking ability which is superior to the RLS and LMS algorithms. Nevertheless, when a projection is performed, noise amplification always arises and this phenomenon degrades the performances of the algorithm. In fact, in the AP algorithm, a covariance matrix of size L is estimated from the data over a rectangular sliding window of size equal to the filter length. In order to apply the projection scheme, this covariance matrix has to be inverted and noise amplification originates from this operation in the case where the covariance matrix is ill-conditioned. This fact is typical in applications where the input signal to the adaptive filter is highly correlated which is the case for speech signals. Several solutions have been given in order to alleviate this problem for Acoustic Echo Cancellation (AEC) applications. In [1], a regularization is achieved by adding δI to the covariance matrix whereas in [2], an exponential window is used in lieu of the rectangular window. All of these methods lead to approximations of the exact AP recursions.

In this paper, we introduce an Instrumental Variable (IV) and derive an algorithm that is situated between the FAP algorithm and the Fast Newton Transversal Filter (FNTF) algorithm. The IV is used in the estimation of a new covariance matrix that replaces the covariance matrix of the AP algorithm. The new covariance matrix appears to be better conditioned. This renders the new algorithm more robust against noise amplification. Nevertheless, the Hermitian structure of the covariance matrix is lost and a new algebraic structure appears. Henceforth, the derivation of a fast algorithm necessitates the derivation of an IV Sliding Window Covariance RLS (IV SWC RLS) algorithm and therefore its fast version which is the IV SWC Fast Transversal Filter (IV SWC FTF) algorithm.

2. THE INSTRUMENTAL VARIABLE AFFINE PROJECTION ALGORITHM

The AP algorithm constitutes a generalization to the Normalized LMS (NLMS) algorithm. For an adaptive filter of length N denoted by $W_{N,k}$ at time k and for an input signal $x(k)$ and the corresponding regression vector $X_N(k) = \left[x^H(k), \cdots, x^H(k-N+1)\right]^H$ (^{H} represents the Hermitian transpose operator), the AP algorithm is given by the following set of equations

$$\epsilon_L^p(k) = d_L(k) + X_{N,L,k} W_{N,k-1}^H \quad (1)$$

$$W_{N,k} = W_{N,k-1} - \mu \epsilon_L^{p^H}(k) R_{L,k}^{-1} X_{N,L,k} \ , \quad (2)$$

with $d_L(k) = [d(k), \cdots, d(k-L+1)]^H$ the vector of the L most recent samples of the desired signal $d(k)$, $\epsilon_L^p(k) =$

Eurecom's research is partially supported by its industrial partners: Ascom, Cegetel, Hitachi, IBM France, Motorola, Swisscom and Thomson CSF.

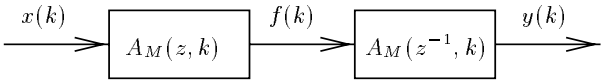

Figure 1: Synthesis of the IV.

$[\epsilon_L(k|k-1)\ \cdots \epsilon_L(k-L+1|k-1)]^H$ is the *a priori* error filtering vector, $\epsilon_L(k|i) = d(k)+W_{N,i}X_N(k)$. $0<\mu<1$ is the step-size (relaxation factor), $X_{N,L,k}$ is the $L\times N$ Hankel data matrix $X_{N,L,k} = [X_N(k)\ \cdots\ X_N(k-L+1)]^H$ and $R_{L,k} = X_{N,L,k}X^H_{N,L,k}$ is the sample covariance matrix estimated on the basis of a rectangular window of length N. The AP algorithm performs at each iteration, a projection of the deviation filter onto the orthogonal subspace to the column space of $X^H_{N,L,k}$.
Consider $y(k)$ to be an IV signal and the corresponding IV regression vector $Y_N(k) = \left[y^H(k)\cdots y^H(k-N+1)\right]^H$, the update equation of the new algorithm is

$$W_{N,k} = W_{N,k-1} - \mu\epsilon^{p\,H}_L(k)\widetilde{R}^{-1}_{L,k}Y_{N,L,k}\ , \qquad (3)$$

with $Y_{N,L,k} = [Y_N(k)\ \cdots\ Y_N(k-L+1)]^H$ and $\widetilde{R}_{L,k} = Y_{N,L,k}X^H_{N,L,k}$. Define the deviation filter to be $\widetilde{W}_{N,k} = W_{N,k} + W^o_N$ with W^o_N denoting the optimal filter and consider the noiseless case where $d_L(k) = X_{N,L,k}W^{o\,H}_N$. From (3), it follows that $\widetilde{W}_{N,k} = \widetilde{W}_{N,k-1}(I - X^H_{N,L,k}\widetilde{R}^{-1}_{L,k}Y_{N,L,k})$, ($\mu = 1$) which reveals the projection scheme of the deviation filter onto the orthogonal subspace to the space column of $X^H_{N,L,k}$ in the direction defined by the row space of $Y_{N,L,k}$.
The IV signal must lead to decorrelation of the input signal. One possible IV is the Kalman gain that is computed in the RLS algorithms. Unfortunately, the Kalman gain does not have the shift invariance property that allows the derivation of fast recursive computations in LS adaptive filtering. Hence, consider the LDU decomposition of the inverse covariance matrix and suppose the prediction filter is of order M (AR(M) assumption) with $M < N$ as is the case for the FNTF algorithm. It appears that the IV signal can be generated according to Fig. (1) where $A_M(z,k) = \sum_{n=0}^{M} A_{n,k}z^{-n}$ and $A_{0,i} = 1$. Note that in the stationary case: $S_{yx}(z) = A_M(z)A_M(z^{-1})S_{xx}(z) = S_{ff}(z)$, hence the cross-covariance matrix R_{yx} is equal to R_{ff} which is diagonal if $A_M(z)$ is the optimal prediction filter associated with $x(k)$. The prediction filter can be time-updated using a trellis structure or an FTF algorithm. $A_M(z^{-1},k)$ being non-causal, it suffices to replace it by $z^{-M}A_M(z^{-1},k)$ and to delay the input and desired signals by M samples to get a realizable structure.

3. THE FAST ALGORITHM

The derivation of the fast algorithm is done in 3 steps. The first step uses the same technique as in [1]. It consists in removing the redundancy in the updating equation (3) due to the successive regression vectors of the data matrix $Y_{N,L,k}$. The second step updates the generators of the inverse covariance matrix. This will be done by deriving the IV SWC FTF algorithm. Finally, the third step, uses the displacement structure of the inverse covariance matrix in order to compute recursively

$$\epsilon^{p\,H}_L(k)\widetilde{R}^{-1}_{L,k} = l^H_L(k) = \left[l^0(k)\cdots l^{(L-1)}(k)\right]\ . \qquad (4)$$

3.1. Fast Computation of the Output Filtering Error

In the first step, the key ingredient is the use of a pseudo-filter $\widehat{W}_{N,k}$ such as

$$W_{N,k} = \widehat{W}_{N,k} - \mu M^H_{L-1}(k)Y_{N,L-1,k}\ , \qquad (5)$$

where

$$M_L(k) = \left[\begin{array}{c} l^{0\,H}(k) \\ \vdots \\ l^{0\,H}(k-L+1) + \cdots + l^{(L-1)\,H}(k) \end{array}\right]\ , \qquad (6)$$

is computed recursively as follows

$$M_L(k) = \left[\begin{array}{c} 0 \\ (M_L(k-1))_{0:L-2} \end{array}\right] + l_L(k)\ . \qquad (7)$$

The pseudo-filter is updated according to

$$\widehat{W}_{N,k} = \widehat{W}_{N,k-1} - \mu Y^H_N(k-L+1)M^{L-1\,H}_L(k)\ , \qquad (8)$$

and the output error filter is computed in the following way

$$\epsilon^p_N(k) = \widehat{\epsilon}^p_N(k) - \mu M^H_{L-1}(k-1)s_{L-1}(k)\ , \qquad (9)$$

where $\widehat{\epsilon}^p_N(k)$ is the output error of the pseudo-filter and

$$\begin{aligned} s_{L-1}(k) \ &= \ Y_{N,L-1,k-1}X_N(k) \ = s_{L-1}(k-1) + \\ & Y^*_{L-1}(k-1)x(k) - Y^*_{L-1}(k-N-1)x(k-N), \end{aligned} \qquad (10)$$

* denoting the complex conjugate operator. The set of equations (5)-(10) constitutes the first step of the fast algorithm we are deriving. In the following, we derive an algorithm that efficiently computes the solution to (4). In order to do this, we have to analyze the displacement structure of $\widetilde{R}^{-1}_{L,k}$. This will be done in an RLS context by first, deriving a recursive version that is the IV SWC RLS algorithm and afterwards, a fast recursive version which is called the IV SWC FTF algorithm.

3.2. The IV SWC RLS Algorithm

When using an IV in the SWC RLS context, one has to verify the orthogonality conditions $\sum_{i=k-N+1}^{k} Y_L(i)\epsilon^H(i|k) = 0_{L\times 1}$ which lead to the normal equations

$$W_{L,N,k}\widetilde{R}_{L,N,k} = -P^H_{L,N,k} \ , \qquad (11)$$

where $\widetilde{R}_{L,N,k} = \widetilde{R}^T_{L,k}$ and $P_{L,N,k} = \sum_{i=k-N+1}^{k} Y_L(i)d_i^H$. Note that in the present case, the indices L and N respectively represent the dimension of the problem and the length of the rectangular window over which the LS solution is computed. We have the following recursions for the sample covariance matrix

$$\begin{aligned} \widetilde{R}_{L,N,k} &= \widetilde{R}_{L,N-1,k-1} + X_L(k)Y_L^H(k) \qquad (12)\\ &= \widetilde{R}_{L,N-1,k} + X_L(k-N+1)Y_L^H(k-N+1) \ , \quad (13)\end{aligned}$$

and for the cross-correlation vector

$$\begin{aligned} P_{L,N,k} &= P_{L,N-1,k-1} + Y_L(k)d^H(k) \qquad (14)\\ &= P_{L,N-1,k} + Y_L(k-N+1)d^H(k-N+1) \ . \quad (15)\end{aligned}$$

The derivation of the recursive version of (11) is done considering two RLS problems: first one being a time and order update recursion $(k-1,N-1) \rightarrow (k,N)$ and the second one is an order down-date recursion $(k,N) \rightarrow (k,N-1)$. The first step is a simple Weighted RLS (exponential window; WRLS) algorithm where the forgetting factor λ is set to 1. Using (12), (13) and the Matrix Inversion Lemma (MIL), it is easy to show that the time and order update part of the recursive algorithm is given by the first set of equations (20). Note that because the Hermitian property of the sample covariance matrix disappeared with the use of the IV, we have to compute 2 Kalman gains $\widetilde{C}_{L,N-1,k}$ and $\widetilde{G}_{L,N-1,k}$ that correspond respectively to the input signal $x(k)$ and to the IV signal $y(k)$. The IV WRLS and its fast version called IV FTF algorithm have been derived in [3]. For the down-date part, let us consider the normal equations $W_{L,N-1,k}\widetilde{R}_{L,N-1,k} = -P^H_{L,N-1,k}$, using (13) and (15), one finds easily

$$W_{L,N-1,k} = W_{L,N,k} + \nu_{L,N-1}(k)F_{L,N,k} \ , \qquad (16)$$

with $F_{L,N,k} = -Y_L^H(k-N+1)\widetilde{R}^{-1}_{L,N,k}$ and

$$\nu_{L,N-1}(k) = d(k-N+1) + W_{L,N-1,k}X_L(k-N+1) \ . \qquad (17)$$

Replacing $W_{L,N-1,k}$ in (17) by the right hand side of (16) gives the following relation

$$\begin{aligned} \nu^s_{L,N}(k) &\triangleq d(k-N+1) + W_{L,N,k}X_L(k-N+1)\\ &= \nu_{L,N-1}(k)\delta^{-1}_{L,N}(k) \ , \qquad (18)\end{aligned}$$

where $\delta^{-1}_{L,N}(k) = 1 - F_{L,N,k}X_N(k-N+1)$. Applying the MIL to (13),we obtain

$$\widetilde{R}^{-1}_{L,N-1,k} = \widetilde{R}^{-1}_{L,N,k} - D^H_{L,N,k}\delta^{-1}_{L,N}(k)F_{L,N,k} \ . \qquad (19)$$

with $D^H_{L,N,k} = -\widetilde{R}^{-1}_{L,N,k}X_N(k-N+1)$. Finally, by associating the time-order update and order down-date equations, the IV SWC RLS algorithm is given by

$$\begin{aligned}
\widetilde{C}_{L,N-1,k} &= -X_L^H(k)\widetilde{R}^{-H}_{L,N-1,k-1}\\
\widetilde{G}_{L,N-1,k} &= -Y_L^H(k)\widetilde{R}^{-1}_{L,N-1,k-1}\\
\gamma_L^{-1}(k) &= 1 - \widetilde{G}_{L,N-1,k}X_L(k)\\
\epsilon^p_{L,N-1}(k) &= d(k) + W_{L,N-1,k-1}X_L(k)\\
\epsilon_{L,N}(k) &= \epsilon^p_{L,N-1}\gamma_N(k)\\
W_{L,N,k} &= W_{L,N-1,k-1} + \epsilon_{L,N}(k)\widetilde{G}_{L,N-1,k}\\
\widetilde{R}^{-1}_{L,N,k} &= \widetilde{R}^{-1}_{L,N-1,k-1} - \widetilde{C}^H_{L,N-1,k}\gamma_N(k)\widetilde{G}_{L,N-1,k}\\
D_{L,N,k} &= -X_N^H(k-N+1)\widetilde{R}^{-H}_{L,N,k} \qquad (20)\\
P_{L,N,k} &= -Y_N^H(k-N+1)\widetilde{R}^{-1}_{L,N,k}\\
\delta_{L,N}(k) &= 1 - P_{L,N,k}X_N(k-N+1)\\
\nu_{L,N}(k) &= d(k-N+1) + W_{L,N,k}X_L(k-N+1)\\
\nu^s_{L,N-1}(k) &= \nu_{L,N}(k)\delta^{-1}_{L,N}(k)\\
W_{L,N-1,k} &= W_{L,N,k} + \nu^s_{L,N-1}(k)P_{L,N,k}\\
\widetilde{R}^{-1}_{L,N-1,k} &= \widetilde{R}^{-1}_{L,N,k} - D^H_{L,N,k}\delta^{-1}_{L,N}(k)P_{L,N,k} \ .
\end{aligned}$$

From, the IV SWC RLS algorithm described above, we can now derive the corresponding fast version called the IV SWC FTF algorithm.

3.3. The IV SWC FTF Algorithm

In what follows, we will only give the equations of the prediction part of IV SWC FTF algorithm. Details about the complete algorithm can be found in [4]. The IV SWC FTF algorithm uses two prediction problems, one associated with the input signal $x(k)$ with $y(k)$ being an IV ($A_{L,N,k}$ and $B_{L,N,k}$ are the corresponding forward and backward prediction filters) and the other associated to the signal $y(k)$ with $x(k)$ as the IV ($\overline{A}_{L,N,k}$ and $\overline{B}_{L,N,k}$ being the forward and backward prediction filters). The prediction part IV SWC FTF algorithm is given by

$$\begin{bmatrix} \widetilde{C}_{L_p,N,k} & \widetilde{G}_{L_p,N,k} \\ A_{L,N,k} & \overline{A}_{L,N,k} \\ \gamma^{-1}_{L_p}(k) & \alpha^{-1}_{L,N}(k) \end{bmatrix} = F_U \begin{bmatrix} \widetilde{C}_{L,N_m,k_m} & \widetilde{G}_{L,N_m,k_m} \\ A_{L,N_m,k_m} & \overline{A}_{L,N_m,k_m} \\ \gamma_L^{-1}(k_m) & \alpha^{-1}_{L,N_m}(k_m) \\ X_{L_p}(k) & Y_{L_p}(k) \end{bmatrix}$$

$$\begin{bmatrix} \widetilde{C}_{L,N_m,k} & \widetilde{G}_{L,N_m,k} \\ B_{L,N,k} & \overline{B}_{L,N,k} \\ \gamma^{-1}_{L}(k) & \beta_{L,N}(k) \end{bmatrix} = F_D \begin{bmatrix} \widetilde{C}_{L_p,N,k} & \widetilde{G}_{L_p,N,k} \\ B_{L,N_m,k_m} & \overline{B}_{L,N_m,k_m} \\ \gamma^{-1}_{L_p}(k) & \beta_{L,N_m}(k_m) \\ X_{L_p}(k) & Y_{L_p}(k) \end{bmatrix}$$

$$\begin{bmatrix} D_{L_p,N_p,k} & P_{L_p,N_p,k} \\ A_{L,N_m,k} & \overline{A}_{L,N_m,k} \\ \delta_{L_p}(k) & \alpha^{-1}_{L,N_m}(k) \end{bmatrix} = F_U \begin{bmatrix} D_{L,N,k_m} & P_{L,N,k_m} \\ A_{L,N,k} & \overline{A}_{L,N,k} \\ \delta_L(k_m) & \alpha^{-1}_{L,N}(k) \\ X_{L_p}(k_N) & Y_{L_p}(k_N) \end{bmatrix}$$

$$\begin{bmatrix} D_{L,N,k} & P_{L,N,k} \\ B_{L,N_m,k} & \overline{B}_{L,N_m,k} \\ \delta_L(k) & \beta_{L,N_m}(k) \end{bmatrix} = F_D \begin{bmatrix} D_{L_p,N_p,k} & P_{L_p,N_p,k} \\ B_{L,N,k} & \overline{B}_{L,N,k} \\ \delta_{L_p}(k) & \beta_{L,N}(k) \\ X_{L_p}(k_N) & Y_{L_p}(k_N) \end{bmatrix} \quad (21)$$

where $L_p = L+1, N_p = N+1, L_m = L-1, N_m = N-1$, $k_m = k-1, k_N = k-N+1$, $\alpha_{L,N}(k)$ and $\beta_{L,N}(k)$ are respectively the forward and backward prediction error energies F_U and F_D are transformations defined in [4]. The computational complexity of the prediction part IV SWC FTF algorithm is $20L$ operations per sample.

3.4. The Complete Algorithm

The last step consists in using the displacement structure of the inverse covariance matrix

$$\widetilde{R}^{-1}_{L,k} = \begin{bmatrix} \widetilde{R}^{-1}_{L_m,k} & 0 \\ 0 & 0 \end{bmatrix} + B^T_{L_m,N,k}\beta^{-1}_{L_m,N,k}(k)\overline{B}^*_{L_m,N,k}$$
$$= \begin{bmatrix} 0 & 0 \\ 0 & \widetilde{R}^{-1}_{L_m,k_m} \end{bmatrix} + A^T_{L_m,N,k}\alpha^{-1}_{L_m,N}(k)\overline{A}^*_{L_m,N,k} \quad (22)$$

Hence, using the fact that $\epsilon^p_L(k) = \left[\epsilon^{p^H}_{L_m}(k) \;\; *\right]^H = \left[\epsilon^{p^*}(k) \;\; (1-\mu)\,(\epsilon^p_L(k_m))^H_{1:L_m}\right]^H$, it is easy to show

$$l^H_L(k) = \begin{bmatrix} 0 & (1-\mu)\psi^H_{L_m}(k_m) \end{bmatrix} + \alpha^{-1}_{L_m,N}(k)A_{L_m,N,k}\epsilon^{p^*}_L(k)\overline{A}^*_{L_m,N,k}\,, \quad (23)$$

$$\begin{bmatrix} \psi_{L_m}(k) \\ 0 \end{bmatrix}^H = l^H_L(k) - \beta^{-1}_{L_m,N}(k)B_{L_m,N,k}\epsilon^{p^*}_L(k)\overline{B}^*_{L_m,N,k}\,.$$

The solution to (4) is computed efficiently with (23) and requires the updating of the forward and backward prediction quantities via the prediction part of the IV SWC FTF algorithm. The generation of the IV needs $2M$ operations for the filtering part and $6M$ operations if one uses the prediction part FTF algorithm for the updating of the prediction filter. Note that if $L > M$ and M is the optimal prediction filter order of the input signal, then one could use the prediction filter computed in the IV SWC FTF algorithm. This allows the generation of the IV in M operations per sample (just filtering the prediction error through $A_M(z^{-1},k)$). Note, that in this case, a prediction part IV SWC FTF algorithm of order M suffices. Combining these results with (5)-(10), we get the IV FAP algorithm. When $L < M$, the computational complexity of the IV FAP algorithm is $2N+28L+8M$ operations per sample for the relaxed version ($\mu \neq 1$) and $2N+22L+8M$ operations per sample for the non-relaxed form ($\mu = 1$). When $L \geq M$, the complexities are respectively $2N+8L+21M$ and $2N+2L+21M$ operations per sample.

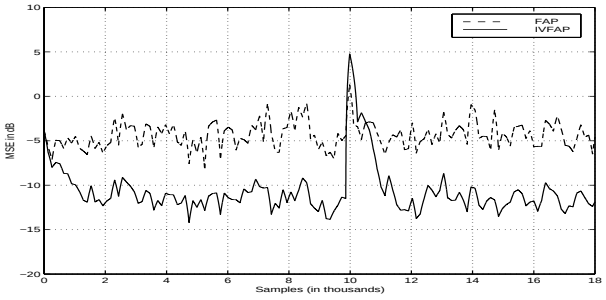

Figure 2: Comparison of the FIVAP and FAP algorithms.

4. SIMULATION

In Fig. 2, we give the learning curves (averaged over 128 samples) of the FIVAP and FAP algorithms for an input which is a highly correlated signal, $N = 256$, $L = 24$, $M = 4$ and $\lambda = .999$ (we used a stabilized FTF with $E_0 = 10$). A sudden variation of the optimal filter arises at $k = 9900$. White output noise has been added so that SNR=20 dB. As one can see, the IVFAP algorithm is more robust to noise amplification than the FAP algorithm.

5. CONCLUDING REMARKS

Due to its numerical error propagation dynamic, the IV SWC FTF algorithm is unstable. One way to overcome this problem is to restart the algorithm whenever instability has been detected. The stabilization of the new algorithm using a feedback mechanism is the subject of our ongoing research.

6. REFERENCES

[1] S. L. Gay and Sanjeev Tavathia. "The Fast Affine Projection Algorithm". In *Proc. ICASSP Conf.*, pages 3023–3026, Detroit, USA, May 9-12 1995.

[2] C.B. Papadias and D.T.M Slock. "New Adaptive Blind Equalization Algorithms for Constant Modulus Constellations". In *Proc. ICASSP*, pages 321–324, Adelaïde, Australia, April. 19-22 1994.

[3] A. Swami. "Fast Transversal Versions of the RIV Algorithm". *International Journal of Adaptive Control and Signal Processing*, 10(2-3):267–281, Mars-June 1996.

[4] Karim Maouche and Dirk T.M. Slock. "The IV SWC FTF algorithm". Technical Report RR N^{O} 37, Institut Eurécom, Sophia Antipolis, France, October 1997.