New Normalized LMS Algorithms Based on the Kalman Filter

Paulo A. C. Lopes and José B. Gerald
IST and INESC-ID, INESC-ID
Rua Alves Redol nº 9
1000-029 Lisboa
Email: paulo.c.lopes@inesc-id.pt

Abstract—While the LMS algorithm and its normalized version (NLMS), have been thoroughly used and studied. Connections between the Kalman filter and the RLS algorithm have been established however, the connection between the Kalman filter and the LMS algorithm has not received much attention. By linking these two algorithms, a new normalized Kalman based LMS (KLMS) algorithm can be derived that has some advantages to the classical one. Their stability is guaranteed since they are a special case of the Kalman filter. More, they suggest a new way to control the step size, that results in good convergence properties for a large range of input signal powers, that occur in many applications. They prevent high measurement noise sensitivity that may occur in the NLMS algorithm for low order filters, like the ones used in OFDM equalization systems. In these paper, different algorithms based on the correlation form, information form and simplified versions of these are presented. The simplified form maintain the good convergence properties of the KLMS with slightly lower computational complexity.

I. INTRODUCTION

The Least Mean Squares (LMS) algorithm for adaptive filters has been extensively studied and tested in a broad range of applications [1]–[4]. In [1] and in [5] a relation between the Recursive Least Squares (RLS) and the Kalman filter [6] algorithm is determined, and in [1] the tracking convergence of the LMS, RLS and extended RLS algorithms, based on the Kalman filter, are compared. However, there is no link established between the Kalman filter and the LMS algorithm.

The classical adaptive filtering problem can be stated in the following manner. Given an reference signal \(u(n)\) and a desired signal \(d(n)\) determine the filter, \(w\), that minimizes the error, \(e(n)\), between the output of the filter, \(y(n)\), and the desired signal, \(d(n)\). For the case of Finite Impulse Responde (FIR) filters, an algorithm that solves this problem is the well known LMS. This is given by,

\[
 w(n + 1) = w(n) + \mu u(n)^* e(n). \tag{1}
\]

This equation updates the vector of the filter coefficients \(w(n)\). The output of the filter is \(y(n) = w^T(n) u(n)\) with \(u(n) = [u(n) \ldots u(n - N + 1)]^T\) where \(N\) is the filter length, and \(e(n) = d(n) - y(n)\).

It is known that the LMS algorithm is only stable if the step size is limited, namely it should be inversely proportional to the power of the reference signal [1]. This leads to the normalized LMS (NLMS) algorithm. It is shown in [1] that this algorithm is stable as long as \(0 < \alpha < 2\) and of course, \(u^T(n) u(n)^* \neq 0\). In order to prevent this last possibility, in practice, the algorithm is usually modified to,

\[
 w(n + 1) = w(n) + \alpha \frac{u(n)^* e(n)}{u^T(n) u(n)^* + q}. \tag{2}
\]

where \(q\) is selected to be small enough when compared with \(u^T(n) u(n)^*\). This is usually chosen in an ad hoc fashion. Techniques to select this value based on the proposed algorithm are presented in the paper.

II. THE KALMAN FILTER

The Kalman filter is based on a state space formulation of a continuous or discrete-time system. We will limit our discussion to discrete time. The system must be linear, but may be time variant. The Kalman filter gives an estimate of the state of the system given a set of outputs. For the case of Gaussian signals, and given the assumed linear model, the state estimate is optimum in the sense that it minimizes the norm of the difference between the estimate and the actual state. The system is described by the equations,

\[
 x(n + 1) = F(n) x(n) + n(n) \tag{3}
\]

\[
 z(n) = H^T(n) x(n) + v(n). \tag{4}
\]

The system state vector is \(x(n)\), and the measured signal vector is given by \(z(n)\). The state transition matrix is \(F(n)\), \(n(n)\) is the state noise, \(H(n)\) is the observation matrix and \(v(n)\) is the measurement noise. The state noise and measurement noise are Gaussian random variables with known autocorrelation functions. The autocorrelation of the state noise is \(Q_{nn}(n)\) and of the measurement noise is \(Q_{vn}(n)\) as in,

\[
 Q_{nn}(n) = E \left[ v(n) v^T(n) \right] \tag{5}
\]

\[
 Q_{vn}(n) = E \left[ n(n) n^T(n) \right]. \tag{6}
\]

The state estimate is given by \(\hat{x}(n)\) in Table I. This table represents the Kalman filter algorithm. The estimate is calculated recursively based on the estimate at the previous time instant, \(\hat{x}(n-1)\). Along with the state estimate, the algorithm updates the state covariance matrix, \(\Sigma_{x(n)}\).
The model for the state variation is,

\[ \lambda w(n+1) = \lambda^2 \Sigma_w(n) + \sigma_w^2(n) + q(n) \]

The resulting algorithm is then,

\[ \alpha(n) = d(n) - u^T(n)w(n) \]

\[ w(n+1) = \lambda w(n) + \frac{\Sigma_w(n)u(n)\alpha(n)}{u^T(n)\Sigma_w(n)u(n) + q(n)} \]

\[ \Sigma_w(n+1) = \lambda^2 \Sigma_w(n) + \frac{\Sigma_w(n)u(n)u^T(n)\Sigma_w(n)}{u^T(n)\Sigma_w(n)u(n) + q(n)} + Q(n) \]

TABLE I

**Kalman Filter**

<table>
<thead>
<tr>
<th>Kalman</th>
<th>Kalman LMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>z(n)</td>
<td>d(n)</td>
</tr>
<tr>
<td>H(n)</td>
<td>u(n)</td>
</tr>
<tr>
<td>x_{i</td>
<td>n-1}</td>
</tr>
<tr>
<td>\Sigma_{x_{i</td>
<td>n-1}}</td>
</tr>
<tr>
<td>Q_{n+1}</td>
<td>Q_{n+1}</td>
</tr>
<tr>
<td>F(n)</td>
<td>q(n)</td>
</tr>
</tbody>
</table>

TABLE II

**Correspondences from the Kalman Filter Variables to Adaptive Filter Variables (NLMS).**

The signal \( \alpha(n) \) is the same as \( e(n) \) and is used to be consistent with the Kalman filter notation. The variance matrix \( \Sigma_w(n) \) can be made diagonal by carefully selecting the state noise autocorrelation matrix \( Q_{n+1}(n) \) at each iteration. More, this can be done without changing the state noise total power, \( \text{tr}\{Q_{n+1}(n)\} \), where \( \text{tr}\{} \) stands for the trace of the matrix. The procedure is not feasible if the state noise is low.

To do this one simply makes \( \Sigma_w(n) = \sigma_w^2(n)I \) and \( \text{tr}\{Q_{n+1}(n)\} = N \sigma_w(n) \) and apply the trace operator to (17). The resulting algorithm is the Kalman based LMS algorithm (KLMS) and is represented in Table III. Note that \( \text{tr}\{u(n)u(n)^T\} = u(n)^T u(n) \). The actual algorithm presented in Table III has been modified to allow complex signals. Namely, in the calculation of the power and in the coefficients update, \( u(n)^* \), the conjugate of \( u(n) \), is used in its place.

**TABLE III**

**Normalized LMS based on the Kalman filter, KLMS**

\[ P(n) = u^T(n)u(n) \]

\[ \alpha(n) = d(n) - u^T(n)w(n) \]

\[ w(n+1) = w(n) + \frac{u(n)^* \alpha(n)}{u^T(n)u(n) + q(n)} \]

\[ \sigma_w^2(n+1) = \sigma_w^2(n) \left( 1 - \frac{P(n)\sigma_w^2(n)}{P(n) + q(n)\sigma_w^2(n)} \right) + q(n) \]

**IV. Choosing the State Noise Variance**

The model for the state variation is,

\[ w_j(n+1) = \lambda w_j(n) + n_j(n). \]

Each coefficient corresponds to a low frequency signal, with time constant given by \( \tau = T/\ln(\lambda) \) where \( T \) is the sampling period. This can be approximated by \( \tau = T/(1 - \lambda) \) if \( \lambda \) is close to one. So one has, \( \lambda \approx (1 - T/\tau) \). The variance of each coefficient is easily calculated as,

\[ \sigma^2_w = \frac{q(n)}{1 - \lambda^2}. \]

This should be equal to the value chosen to initialize the algorithm \( \sigma_w^2 = \sigma_{w,0}^2 \). It follows that the state noise can be chosen as,

\[ q(n) = \sigma_w^2(1 - \lambda^2) \approx 2\sigma_{w,0}^2 \frac{T}{\tau} \]

where the last approximation is valid for large \( \tau \), where \( \tau \) is the time constant of the underlaying model, as previously discussed.
V. INFORMATION FORM KALMAN BASED LMS ALGORITHM

If the state noise is low or zero (17) can be written as,

\[ \Sigma_w^{-1}(n+1) = \Sigma_w^{-1}(n) + \frac{u(n)u^T(n)}{q_v(n)}. \]  

(27)

The matrix \( \Sigma_w^{-1}(n+1) \) can be approximated by a diagonal matrix if the reference signal autocorrelation is narrow. Doing this and applying the trace operator results,

\[ \sigma_w^{-2}(n+1) = \sigma_w^{-2}(n) + \frac{P(n)}{N q_v(n)}. \]  

(28)

by defining the total power up to (but not counting) time \( n \), \( P_T(n) \), by the equation,

\[ P_T(n+1) = P_T(n) + P(n) \]  

(29)

one can prove that,

\[ \sigma_w^{-2}(n) = \frac{P_T(n) + N q_v(n^2)}{N q_v(n)} \]  

(30)

with, \( P_T(0) = 0 \), resulting in the algorithm presented in Table IV. Note that this algorithm is equivalent to the KLMS for the case \( N = 1 \).

---

| Initialize | \( w(0) = w_0 \) | (31) |
| P\(_T(0) = 0\) | (32) |

---

Iterate from \( n = 0 \) to ...

\[ \alpha(n) = d(n) - u^T(n)w(n) \]  

(33)

\[ P(n) = u(n)^T u(n) \]  

(34)

\[ w(n+1) = w(n) + \frac{u(n)^* \alpha(n)}{P(n) + P_T(n)/N + q_v/\sigma_w^2} \]  

(35)

\[ P_T(n+1) = P_T(n) + P(n) \]  

(36)

---

Table IV

INFORMATION FORM KALMAN BASED LMS (IKLMS)

VI. SIMPLIFICATIONS OF THE ALGORITHMS

If one is not interested in the initial convergence, then the algorithm in Table III can be simplified. The coefficients estimation error standard deviation \( \sigma_w^2(n) \) converges to a steady state value, resulting that \( q_v(n)/\sigma_w^2(n) \) converges to,

\[ q_v(\infty)/\sigma_w^2(\infty) = \frac{P}{2} \left( -1 + \sqrt{1 + \frac{4 q_v}{N P q_n}} \right). \]  

(37)

This can be used in place of \( q_v(n)/\sigma_w^2(n) \). The value of the state noise can be calculated as in (26).

The algorithm in IV can also be simplified. The time varying quantity \( P_T(n+1) \) is replaced by an estimate of its value at time \( M \), resulting in,

\[ w(n+1) = w(n) + \frac{u(n)^* \alpha(n)}{(N+M-1) u^T(n) u(n) + q_v/\sigma_w^2} \]  

(38)

We call this algorithm the Simplified Information Form Kalman LMS (SIKLMS). The quantity \( M \) is the step sample time.

VII. SENSITIVITY TO MEASUREMENT NOISE

The use of the NLMS algorithm can lead to amplification of the measurement noise in low order filters when the reference signal power takes low values. This can be seen by assuming \( d(n) = u^T(n)w_{op}(n) + v(n) \) and rearranging the NLMS algorithm to,

\[ w(n+1) = (I - \Gamma) w(n) + \frac{v(n)}{u^T(n) u(n)} \]  

(39)

where \( \Gamma \) is a matrix given by,

\[ \Gamma = \frac{u(n)^* u(n)^T}{u^T(n) u(n) + q_v}. \]  

(40)

Equation (39) can be made diagonal to represent a bank of low-pass first order IIR filters with added noise, the last term in the equation. For low reference signal power this term will assume high values, resulting in poor performance of the algorithm. The KLMS solves this problem by carefully selection of the value of \( q \).

VIII. SIMULATION RESULTS

Simulation results are presented for the case of a one-coefficient complex-filter and a ten-real-coefficient filter. Comparisons are made with the LMS and NLMS algorithm. The one-coefficient complex filter is typically used in orthogonal frequency division multiplexing (OFDM) channel equalization. In this application, equalization is done in the frequency domain resulting in complex filters. Also, due to the presence of nulls in the channel frequency response and due to the low-pass characteristics of many channels, the input signal power varies considerably.

The measurement noise, which is prior to the algorithm, can be considered constant, resulting in a large variation of the sinal-to-noise ratio (SNR). This fits nicely to the KLMS formulation. Also, the NLMS will perform poorly when the input signal power takes low values, as shown in the simulations.

Fig. 1 presents the convergence curves of the mean square error between the output of the adaptive filter and the desired signal for the case of a one-coefficient complex filter for channel equalization and tracking. The reference signal is the output of the channel and the desired signal is the input of the channel. The input of the channel was a QAM64 signal with a power of one. A noise signal with standard deviation of 0.25 was added at the output of the channel. This means that the channel is driven with a capacity gap of 3 dB. The measurement noise power of the KLMS has set to, \( q_v = (0.25)^2 \), the optimal value. The step size of the LMS and NLMS and the step sample time, \( M \), of the SIKLMS were optimized to maximize the convergence rate of the algorithms, resulting in the values of 0.5, 0.5 and 2.0. The curves are the result of the ensemble average of 100 trials.
It can be seen that the KLMS has the best performance. In the case of the NLMS, due to the low filter order, occasional low values of the reference signal power result in very high values of the residual error. The LMS and SIKLMS both have good results.

In Fig. 2, the reference signal level was amplified by 40%, while the parameters of all the algorithms were kept constant. It can be seen that the LMS algorithm gets unstable. The NLMS has fewer problems, but it still suffers from measurement noise amplification occasionally. The KLMS and SIKLMS still give good results.

Fig. 3 provides a comparison of the convergence of the mean square error of the NLMS, KLMS, IKLMS and SIKLMS. The curves of the KLMS, SIKLMS and NLMS are very similar. The desired signal was equal to the reference signal filtered by a bandpass filter, with unit gain at the center frequency. The reference signal had unit power and the RMS of the measurement error was 0.03. The adaptive algorithms were only started after ten iterations, when the input buffer was full. The step size of the NLMS and SIKLMS was optimized for best performance while in the KLMS and IKLMS the algorithms parameters were chosen naturally. The Kalman filter is, of course, the fastest but was a much higher computational cost. The IKLMS is the slowest.

IX. CONCLUSION

A new version of the NLMS algorithm based on the Kalman filter (the KLMS) was derived. The new algorithm is stable since, it was derived from the Kalman filter. It allows faster convergence and much higher noise immunity when the reference signal vector norm takes low values, namely in the case of low order filters (like in OFDM systems). In the NLMS algorithm, $q_i$ prevents division by zero. In the KLMS accurate formulas for $q$ give it good noise immunity properties. The SIKLMS achieves close performance at slightly lower computational cost.

REFERENCES