

# Sobre el algoritmo acelerador para filtraje adaptativo

## Resumen

El algoritmo Acelerador para filtraje adaptativo presenta un compromiso más favorable entre la velocidad de convergencia y la variancia de la estimativa cuando es comparado con los algoritmos LMS o NLMS. Eso se consigue a costa de un moderado aumento en la complejidad computacional. En este trabajo el algoritmo Acelerador es interpretado como siendo del tipo cuasi-Newton, además también se presenta una versión simplificada de ajuste más simple. Para esta versión se presenta un análisis de los momentos de primera y segunda ordenes que resulta en expresiones para el error cuadrático medio y el desajuste. Usando estas expresiones, una comparación simple con el algoritmo LMS muestra las ventajas potenciales del algoritmo Acelerador. También son presentados resultados de simulaciones.

## Introducción

**E**n (Gerken, 2000) fue introducido un algoritmo rápido de filtraje adaptativo de tiempo discreto, derivado de un sintonizador de tiempo continuo que fija la segunda derivada (aceleración) de la estimativa de los parámetros (Pait, 1968). En principio tal sintonizador tiene aplicaciones en control adaptativo. Mediante simulaciones fue mostrado en (Gerken, 2000) que, para señales de entrada coloridas, el algoritmo propuesto puede presentar en régimen un compromiso más favorable entre rapidez de convergencia y error de estimación que los algoritmos LMS y NLMS, una propiedad obtenida a costas de un

moderado aumento en la complejidad computacional. Fue también presentada una demostración de la convergencia determinística.

Como ya fue mencionado, el algoritmo Acelerador (AC) está basado en un sintonizador de tiempo continuo que ajusta la segunda derivada, con respecto al tiempo, de los coeficientes del filtro. Igualmente, el popular algoritmo LMS puede ser obtenido de un sintonizador de tiempo continuo que ajusta la primera derivada, con respecto al tiempo, de los coeficientes del filtro. Por otro lado, en el algoritmo LMS se utiliza una estimativa de la primera derivada, con respecto a los coeficientes del filtro, de una función costo. Así se puede sospechar que en el algoritmo AC se hace uso de una

estimativa de la segunda derivada con respecto a los coeficientes del filtro. De hecho, esta sospecha es correcta y en este artículo es mostrado que el algoritmo AC puede ser interpretado como un método cuasi-Newton, siendo presentadas las aproximaciones del vector gradiente y de la matriz Hessiana utilizadas.

En su forma general el algoritmo AC depende de cuatro parámetros, un escalar y tres matrices positivas definidas, que deben ser seleccionados adecuadamente. Se por un lado os parámetros extras adicionan libertad al proyectista del filtro adaptativo, por otro lado a selección de ellos es una dificultad en el uso del algoritmo AC, lo que motiva a obtención de nuevas versiones que sean más fáciles de ajustar. En este artículo es presentada una versión simplificada del algoritmo AC y los resultados de un análisis de los momentos de primera y segunda ordenes de los errores de los coeficientes del filtro. Con eso son obtenidas expresiones que caracterizan el comportamiento transitorio y en régimen del error cuadrático medio y el desajuste. Usando estas expresiones una simple comparación con el algoritmo LMS es presentada, mostrando las posibles ventajas de usar el algoritmo AC. Por ejemplo, se muestra que para velocidades de convergencia semejantes el algoritmo AC presenta un menor desajuste que el algoritmo LMS. Esta "atenuación" del desajuste puede ser tanto mayor cuanto más mal acondicionadas sean las señales de entrada. Como resultado de la comparación de desempeño con el bien conocido algoritmo LMS, se obtienen algunas indicaciones de cuño práctico para seleccionar los parámetros del algoritmo AC.

En el texto a seguir es inicialmente presentada una versión simplificada del algoritmo AC. En seguida es mostrado que el algoritmo AC, tanto en la versión completa como en la simplificada, puede ser considerado como siendo del tipo cuasi-Newton. Posteriormente son presentados algunos resultados del análisis de los momentos de primera y segunda ordenes y la comparación con el algoritmo LMS. Siempre que sea apropiado son presentados algunos resultados de simulaciones. Se finaliza el artículo con una sección de conclusiones.

## Una versión simplificada del algoritmo acelerador

De la misma forma que otros algoritmos de filtraje adaptativo, el algoritmo AC ajusta recursivamente una estimativa  $p[n]$  de un vector de coeficientes po usando una señal deseada  $d[n] = x^T[n]p_0 - \eta[n]$ , con  $x[n]$  siendo el vector de señal de entrada (regresor) y  $\eta[n]$  el ruido de medida. El objetivo es mantener el error de estimación a posteriori  $e[n] = x^T[n]p[n] - d[n]$ , el error de estimación a priori  $e_a[n] = x^T[n]p[n-1] - d[n]$ , y el error de los coeficientes  $\delta p[n] = p[n] - p_0$  tan pequeños cuanto sea posible.

La forma original del algoritmo AC es dado por [1]:

$$e_a[n] = x^T[n]p[n-1] - d[n] \quad (1)$$

$$G[n] = I + \alpha^2 M_1 x[n]x^T[n] + 2\alpha M_1 (M_2 + x[n]x^T[n]M_1 M_3) \quad (2)$$

$$q[n] = (G[n])^{-1} (q[n-1] - \alpha M_1 x[n]e_a[n]) \quad (3)$$

$$p[n] = p[n-1] + \alpha q[n]. \quad (4)$$

En las ecuaciones anteriores  $\alpha$  es un escalar positivo,  $I$  es la matriz identidad y  $M_i$ ,  $i = 1; 2, 3$ , son matrices simétricas positivas definidas que deben satisfacer las siguientes desigualdades para garantizar la estabilidad del algoritmo:

$$4M_1 M_3 M_1 \geq M_2^{-1} \quad (5)$$

$$M_2 M_1 M_3 + M_3 M_1 M_2 \geq M_1^{-1}/2. \quad (6)$$

En [1] fue mostrado que la actualización (3) de  $q[n]$  puede ser hecha de la siguiente forma:

$$C[n] = \frac{\alpha e_a[n] + x^T[n] B M_1^{-1} q[n-1]}{1 + x^T[n] B x[n]} M_1$$

$$q[n] = A (q[n-1] - C[n] x[n]),$$

con

$$A = (I + 2\alpha M_1 M_2)^{-1},$$

$$B = \alpha (\alpha I + 2M_1 M_3) A M_1.$$

Motivado por el hecho de que la mayor rapidez de convergencia se obtiene en el límite superior de (5) y (6), son introducidas las relaciones  $M_1 M_2 = I/(2\gamma)$  y  $M_1 M_3 = \gamma I/2$ . Así, para un escalar positivo  $\gamma$  y una matriz definida positiva  $M_1$  los límites superiores

de (5) y (6) son siempre conseguidos. Con estas consideraciones se obtiene una versión simplificada del algoritmo AC que es dada por:

$$\begin{aligned} e_a[n] &= \mathbf{x}^T[n] \mathbf{p}[n-1] - d[n] \\ g[n] &= \frac{e_a[n] + \gamma \mathbf{x}^T[n] \mathbf{q}[n-1]}{1 + \alpha \gamma \mathbf{x}^T[n] \mathbf{M}_1 \mathbf{x}[n]} \\ \mathbf{q}[n] &= \frac{\gamma}{\alpha + \gamma} (\mathbf{q}[n-1] - \alpha g[n] \mathbf{M}_1 \mathbf{x}[n]) \\ \mathbf{p}[n] &= \mathbf{p}[n-1] + \alpha \mathbf{q}[n]. \end{aligned} \quad (7)$$

Una simplificación adicional puede ser introducida imponiendo  $\mathbf{M}_1 = m_1 \mathbf{I}$ , siendo  $\mathbf{I}$  la matriz identidad. Se debe recordar que, en conformidad con los resultados ya mencionados, el algoritmo anterior será estable para parámetros  $\alpha$ ,  $\gamma$  y  $m_1$  positivos (o  $\mathbf{M}_1$  positiva definida).

### El algoritmo acelerador y el método de Newton

Dada una función a ser minimizada  $f(\mathbf{p})$ , en el método de Newton se aproxima esta función localmente por una función cuadrática y se determina el mínimo de la aproximación y no de la función original (Luenberger, 1989). Cerca de  $\mathbf{p}_{n-1} = \mathbf{p}[n-1]$  podemos aproximar  $f(\mathbf{p})$  pela serie truncada de Taylor

$$f(\mathbf{p}) \simeq f(\mathbf{p}_{n-1}) + \nabla f^T(n, \mathbf{p}_{n-1}) (\mathbf{p} - \mathbf{p}_{n-1}) + \frac{1}{2} (\mathbf{p} - \mathbf{p}_{n-1})^T \mathbf{F}(n, \mathbf{p}_{n-1}) (\mathbf{p} - \mathbf{p}_{n-1}), \quad (8)$$

siendo  $\nabla f(\cdot; \cdot)$  el vector gradiente definido como un vector columna  $M$ -dimensional y  $\mathbf{F}(\cdot; \cdot)$  la matriz Hessiana de  $f(\mathbf{p})$ . En el algoritmo de Newton  $\mathbf{p}[n]$  es ajustado para minimizar el lado derecho de (8) de la siguiente forma:

$$\mathbf{p}_n = \mathbf{p}_{n-1} - \mathbf{F}^{-1}(n, \mathbf{p}_{n-1}) \nabla f(n, \mathbf{p}_{n-1}). \quad (9)$$

Es ilustrativo considerar a minimización de la función costo cuadrática  $f(\mathbf{p}) = E\{e^2[n]\}/2$ , siendo  $E\{\cdot\}$  el operador esperanza y  $e[n] = \mathbf{x}^T[n] \mathbf{p} - d[n]$ . Considerando una aproximación como en (8) se obtiene:

$$\mathbf{F}(n, \mathbf{p}_{n-1}) = \mathbf{R}_x = E\{\mathbf{x}[n] \mathbf{x}^T[n]\} \quad (10)$$

$$\nabla f(n, \mathbf{p}_{n-1}) = \mathbf{R}_x \mathbf{p}[n-1] - E\{d[n] \mathbf{x}[n]\} \quad (11)$$

Substituyendo estas ecuaciones en (9) resulta

$$\mathbf{p}[n] = \mathbf{R}_x^{-1} E\{d[n] \mathbf{x}[n]\},$$

observándose de este modo que en este caso el algoritmo de Newton converge en una única iteración.

Aun cuando las propiedades de convergencia del método de Newton sean muy atractivas, el cálculo de la inversa de la matriz Hessiana no es adecuado para una implementación práctica. La idea básica de los métodos cuasi-Newton es usar aproximaciones de la inversa de la matriz Hessiana. La primera modificación en la dirección de un método cuasi-Newton es usualmente la introducción de un paso de adaptación  $\rho_n$ :

$$\mathbf{p}_n = \mathbf{p}_{n-1} - \rho_n \mathbf{F}^{-1}(n, \mathbf{p}_{n-1}) \nabla f(n, \mathbf{p}_{n-1}). \quad (12)$$

Para verificar que el algoritmo AC puede colocarse en la forma (12) utilizando aproximaciones para a inversa de la matriz Hessiana y para el vector gradiente basta combinar las expresiones (3) y (4). Así se obtiene:

$$\begin{aligned} \mathbf{p}[n] &= \mathbf{p}[n-1] + \alpha (\mathbf{G}[n])^{-1} \{\mathbf{q}[n-1] - \alpha \mathbf{M}_1 \mathbf{x}[n] e_a[n]\} \\ &= \mathbf{p}[n-1] - (\mathbf{G}[n])^{-1} \{\alpha^2 \mathbf{M}_1 \mathbf{x}[n] e_a[n] + \\ &\quad - (\mathbf{p}[n-1] - \mathbf{p}[n-2])\}. \end{aligned} \quad (13)$$

Comparando (13) con (12) se puede concluir inmediatamente que el algoritmo AC hace uso das siguientes estimativas:

$$\widehat{\nabla} f(n, \mathbf{p}_{n-1}) = \beta (\alpha^2 \mathbf{M}_1 e_a[n] \mathbf{x}[n] + \mathbf{p}[n-1] - \mathbf{p}[n-2]) \quad (14)$$

$$\widehat{\mathbf{F}}(n, \mathbf{p}_{n-1}) = \beta \mathbf{G}[n], \quad (15)$$

siendo  $\beta$  una constante positiva adecuada. Considerando la versión simplificada con  $\mathbf{M}_1 = m_1 \mathbf{I}$  y  $\beta = \gamma/(\alpha + \gamma)$  resultan expresiones más familiares:

$$\rho_n = \frac{\alpha}{\alpha + \gamma} \alpha \gamma m_1 \quad (16)$$

$$\widehat{\nabla} f(n, \mathbf{p}_{n-1}) = e_a[n] \mathbf{x}[n] + \frac{1}{\alpha^2 m_1} (\mathbf{p}[n-1] - \mathbf{p}[n-2]) \quad (17)$$

$$\widehat{\mathbf{F}}(n, \mathbf{p}_{n-1}) = \mathbf{I} + \alpha \gamma m_1 \mathbf{x}[n] \mathbf{x}[n]^T. \quad (18)$$

La expresión (17) es una estimativa instantánea del vector gradiente  $E\{e_a[n]x[n]\}$  "corregida" por un término proporcional al incremento del coeficiente anterior, y (18) es una estimativa instantánea regularizada de la matriz Hessiana  $Rx = E\{x[n]x^T[n]\}$ . Es ilustrativo observar que para el algoritmo LMS vale  $\rho_n = \mu$  (o paso de adaptación),

$$\widehat{\nabla} f(n, \mathbf{p}_{n-1}) = e_a[n]x[n] \text{ e } \widehat{\mathbf{F}}(n, \mathbf{p}_{n-1}) = \mathbf{I}.$$

Para ilustrar la naturaleza Newton del algoritmo AC será considerado un ejemplo simple de un filtro predictor de dos coeficientes excitado por un señal sinusoidal  $\sin((5\pi/128)n + 0;4\pi)$  y sujeto a las siguientes condiciones: coeficientes iniciales -1;5 y -1;7 y ruido de medida blanco, Gaussiano, con variancia  $10^{-4}$ . Los coeficientes óptimos son 2 y -1. Son considerados tres algoritmos: LMS ( $\mu = 0;03$ ), Newton ( $\rho_n = 0;05$ ; ver Ec. (10), (11) y (12)) y AC ( $\alpha = 0;125$ ;  $m_1 = 0;09$ ;  $\gamma = 15$  y  $\gamma = 320$ ). Los pasos de adaptación y los parámetros de todos los algoritmos, excepto el de Newton, fueron ajustados para producir la misma desviación estandar de los coeficientes (0;002) después de la convergencia. La Fig. 1 presenta las trayectorias en el plano de los coeficientes donde también están representadas las curvas de contorno de la función costo. Dos comportamientos diferentes del algoritmo AC son mostrados, uno de ellos similar al del algoritmo LMS (AC1 con  $\gamma = 15$ ) y otro similar al algoritmo de Newton (AC2 con  $\gamma = 320$ ). En este último caso, considerando el espacio de los coeficientes, la convergencia ocurre aproximadamente sobre una reta ligando los coeficientes iniciales a los coeficientes óptimos. Queda por tanto evidenciado que, dependiendo del valor de los parámetros  $\alpha$ ,  $m_1$  y  $\gamma$ , el algoritmo acelerador puede presentar comportamientos bastante diversos. Puede causar una cierta sorpresa que cuando el algoritmo AC presenta un comportamiento como el algoritmo de Newton, la velocidad de convergencia es menor de que cuando el comportamiento es semejante al del algoritmo LMS. Una posible explicación para este comportamiento es que los parámetros del algoritmo AC deben estar ajustados de tal forma que un mayor número de iteraciones ocurra para que buenas estimativas del gradiente y de la inversa da matriz Hessiana sean obtenidas.

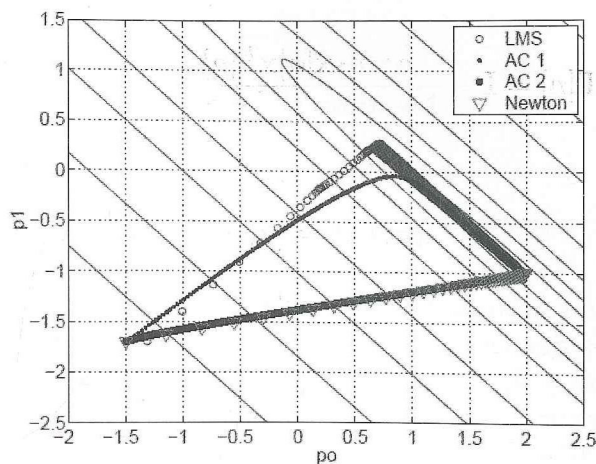


Fig. 1. Trayectorias en el plano de los coeficientes para un predictor con dos coeficientes. AC1:  $\gamma = 15$ , AC2:  $\gamma = 320$ .

Cerrando esta sección sigue un resumen de las principales características del algoritmo AC tratadas hasta ahora:

- (i) El algoritmo AC minimiza la función  $E\{e^2[n]\}$ .
- (ii) En cada paso se considera una aproximación local de la función costo como la dada en (8) usando estimativas no triviales para el vector gradiente y para la matriz Hessiana, dadas respectivamente por (14) y (15), o por (17) y (18) para la versión simplificada.
- (iii) En el sentido de (i) y (ii) el algoritmo AC puede clasificarse como un método cuasi-Newton.
- (iv) Conforme es mostrado en (Gerken, 2000) el algoritmo AC es estable, lo que significa que con las estimativas utilizadas del vector gradiente y de la matriz Hessiana la convergencia queda asegurada.

### Análisis de los momentos de primera y segunda órdenes.

Observando que  $d[n] = x^T[n]p_0 - \eta[n]$  y  $\delta p[n] = p[n] - p_0$  se puede escribir:

$$e_a[n] = x^T[n]p[n] - d[n] \\ = x^T[n]p[n] + \eta[n],$$

con  $\eta[n]$  siendo el ruido de medida. Introduciendo la variable  $r[n] = \delta p[n] + \gamma q[n]$  en las Ecuaciones (7) resulta:

$$r[n] = \Gamma[n]r[n-1] + \alpha \gamma m_1 \Gamma[n]x[n]\eta[n] \quad (19)$$

$$\delta p[n] = a\delta p[n-1] + br[n], \quad (20)$$

con

$$\Gamma[n] = \mathbf{I} - \frac{\alpha\gamma m_1 \mathbf{x}[n] \mathbf{x}^T[n]}{1 + \alpha\gamma m_1 \mathbf{x}^T[n] \mathbf{x}[n]}$$

y

$$a = 1 - b = \gamma / (\hat{a} + \gamma).$$

Con estas ecuaciones se puede realizar un análisis de los momentos de primera y segunda ordenes de  $\mathbf{r}[n]$  y  $\delta\mathbf{p}[n]$ , semejante al análisis usual de los momentos del algoritmo LMS (Solo y Kong, 1989; Haykin, 1996). Para eso se hace uso de las siguientes suposiciones simplificadoras:

-  $\mathbf{x}[n]$  es un vector de ruido blanco Gaussiano de media nula con matriz de covariancia

$$\mathbf{R}_x = E\{\mathbf{x}[n] \mathbf{x}^T[n]\}.$$

-  $\eta[n]$  es un ruido blanco Gaussiano de media nula y variancia  $\sigma_\eta^2$  y estadísticamente independiente de  $\mathbf{x}[n]$ .

- La orden  $M$  del filtro adaptativo es grande lo suficiente para hacer la variable  $\mathbf{x}^T[n] \mathbf{x}[n]$  se comporte de forma "lenta" en comparación con  $\mathbf{x}[n]$ .

Esta última suposición posibilita la aplicación de la propiedad de las medias ("averaging property") [5] para obtener

$$\begin{aligned} E\{\Gamma[n]\} &\approx \mathbf{I} - \frac{\alpha\gamma m_1 E\{\mathbf{x}[n] \mathbf{x}^T[n]\}}{1 + \alpha\gamma m_1 E\{\mathbf{x}^T[n] \mathbf{x}[n]\}} \\ &= \mathbf{I} - \frac{\alpha\gamma m_1}{1 + \alpha\gamma m_1 \text{tr}\{\mathbf{R}_x\}} \mathbf{R}_x = \mathbf{I} - \mu_a \mathbf{R}_x, \end{aligned} \quad (21)$$

con  $\text{tr}\{\cdot\}$  representando el trazo de la matriz y

$$\mu_a = \frac{\alpha\gamma m_1}{1 + \alpha\gamma m_1 \text{tr}\{\mathbf{R}_x\}}. \quad (22)$$

En estas condiciones es posible mostrar que:

- Los momentos de primera orden de  $\mathbf{r}[n]$  y  $\delta\mathbf{p}[n]$  satisfacen

$$E\{\mathbf{r}[n]\} = (\mathbf{I} - \mu_a \mathbf{R}_x) E\{\mathbf{r}[n-1]\} \quad (23)$$

$$E\{\delta\mathbf{p}[n]\} = a E\{\delta\mathbf{p}[n-1]\} + b E\{\mathbf{r}[n]\} \quad (24)$$

- Las matrices de covariancia  $\mathbf{R}[n] = E\{\mathbf{r}[n] \mathbf{r}^T[n]\}$ ,  $\mathbf{P}[n] = E\{\delta\mathbf{p}[n] \delta\mathbf{p}^T[n]\}$  y de covariancia cruzada  $\mathbf{V}[n] = E\{\delta\mathbf{p}[n-1] \mathbf{r}^T[n]\}$  satisfacen

$$\begin{aligned} \mathbf{R}[n] &= (\mathbf{I} - \mu_a \mathbf{R}_x) \mathbf{R}[n-1] (\mathbf{I} - \mu_a \mathbf{R}_x) \\ &\quad + \mu_a^2 \mathbf{R}_x \sigma_\eta^2 \end{aligned} \quad (25)$$

$$\begin{aligned} \mathbf{V}[n] &= a \mathbf{V}[n-1] (\mathbf{I} - \mu_a \mathbf{R}_x) \\ &\quad + b \mathbf{R}[n-1] (\mathbf{I} - \mu_a \mathbf{R}_x) \end{aligned} \quad (26)$$

$$\begin{aligned} \mathbf{P}[n] &= a^2 \mathbf{P}[n-1] + b^2 \mathbf{R}[n] + ab \mathbf{V}[n] \\ &\quad + ab \mathbf{V}^T[n] \end{aligned} \quad (27)$$

- El error cuadrático medio  $E\{e_a^2[n]\}$  satisface

$$E\{e_a^2[n]\} = \text{tr}\{\mathbf{R}_x \mathbf{P}[n-1]\} + \sigma_\eta^2. \quad (28)$$

Con  $n \rightarrow \infty$  se obtienen de las ecuaciones (23) a (28) el trazo de la matriz de covariancia (en régimen) de los errores dos coeficientes

$$\text{tr}\{\mathbf{P}_\infty\} = \sigma_\eta^2 \frac{\alpha}{\alpha + 2\gamma} \sum_{k=1}^M \frac{\mu_a}{2 - \mu_a \lambda_k} g(\mu_a \lambda_k) \quad (29)$$

y el error cuadrático medio de estimación después de la convergencia

$$E\{e_a^2\} = \sigma_\eta^2 + \underbrace{\sigma_\eta^2 \frac{\alpha}{\alpha + 2\gamma} \sum_{k=1}^M \frac{\mu_a \lambda_k}{2 - \mu_a \lambda_k} g(\mu_a \lambda_k)}_{\mathcal{M}_a}, \quad (30)$$

con  $\lambda_k$  siendo los auto valores de  $\mathbf{R}_x$  y

$$g(\mu_a \lambda_k) = 1 + \frac{2(1 - \mu_a \lambda_k)}{\alpha/\gamma + \mu_a \lambda_k}. \quad (31)$$

Observándose las Ecuaciones (23) y (25) se concluye que en el caso del algoritmo AC la constante  $\mu_a$  hace el mismo papel del paso de adaptación  $\mu$  del algoritmo LMS (ver Cap. 9 de Haykin, 1996). Ese hecho lleva a la denominación de "paso de adaptación equivalente" para la constante  $\mu_a$ .

Considerando (31) se observa que

$$g(\mu_a \lambda_k) = 1 + \frac{2(1 - \mu_a \lambda_k)}{\alpha/\gamma + \mu_a \lambda_k} < \frac{\alpha + 2\gamma}{\alpha}$$

pues  $\mu_a \lambda_k \geq 0$ . Así, para un paso de adaptación equivalente  $\mu_a$  igual al paso de adaptación del algoritmo LMS ( $\mu = \mu_a$ ), resulta de (30) que el desajuste del algoritmo AC ( $M_a$ ) es siempre menor que el desajuste del algoritmo LMS ( $M_{lms}$ ) (ver Cap. 9 de Haykin, 1996). Por otro lado, la matriz de covariancia del error de los coeficientes del algoritmo LMS satisface la misma ecuación de diferencias que  $R[n]$ . Por lo tanto, desde que las desigualdades

$$\begin{aligned} \gamma/(\alpha + \gamma) &< 1 - \mu_a \lambda_{\min} \\ \alpha/(\alpha + \gamma) &> \mu_a \lambda_{\min} \end{aligned} \quad (32)$$

sean satisfechas,  $P[n]$  en (27) tiene esencialmente el mismo comportamiento de convergencia que  $R[n]$  en (25). De este modo, con esta condición satisfecha, se puede esperar que el algoritmo AC tenga aproximadamente la misma rapidez de convergencia que el LMS con paso de adaptación  $\mu = \mu_a$ . Recordando que para el algoritmo LMS  $\mu \lambda_{\min} < 2\lambda_{\min}/\lambda_{\max}$ , se concluye que la atenuación del desajuste del AC sobre el algoritmo LMS puede ser aumentada conforme la matriz  $R_x$  sea más mal condicionada manteniéndose a rapidez de convergencia constante.

Para verificar hasta donde el raciocinio anterior es válido fueron realizadas algunas simulaciones. Fue considerada la identificación de un sistema FIR con  $M = 6$  coeficientes iguales a 1. Como señal de entrada fue tomada una señal Gaussiana colorida de media nula y variancia  $\sigma_x^2$ . Como ruido de medida fue considerado un ruido Gaussiano blanco de media nula y variancia  $\sigma_\eta^2$ . Los valores utilizados están relacionados en la Tabla 1.

En la Fig. 2 es mostrada la influencia de  $\gamma/(\alpha + \gamma)$  sobre la rapidez de convergencia del algoritmo AC. Los correspondientes valores de los parámetros son presentados en la Tabla 1. La comparación de los valores de  $\gamma/(\alpha + \gamma)$  y  $1 - \mu \lambda_{\min}$  y las curvas del error cuadrático medio correspondientes confirman el comportamiento esperado.

$m_1$	$\gamma$	$\frac{\gamma}{\alpha + \gamma}$	$1 - \mu \lambda_{\min}$	Fig.2
60	0,01	0,242	0,987	AC 1
6	0,1	0,762	0,987	AC 2
0,6	1	0,970	0,987	AC 3
$\sigma_x^2 = 4, \sigma_\eta^2 = 0,5, \alpha = 0,0313, \mu_a = \mu = 0,0129$				

Tabla 1. Parámetros de la Fig. 2.

En la Tabla 2 son comparados los desajustes medidos y los desajustes teóricos de los algoritmos AC y LMS para algunas situaciones. Comparándose los desajustes medidos de los algoritmos LMS y AC se observa que la reducción de desajuste alcanzado por el algoritmo AC es de aproximadamente 25%. Además, comparándose los desajustes teórico y medido del algoritmo AC se observa que los valores medidos son menores que los teóricos. Aun cuando los valores medidos presenten un comportamiento cualitativo correcto hay un desvío entre los resultados del modelo obtenido y los resultados de las simulaciones.

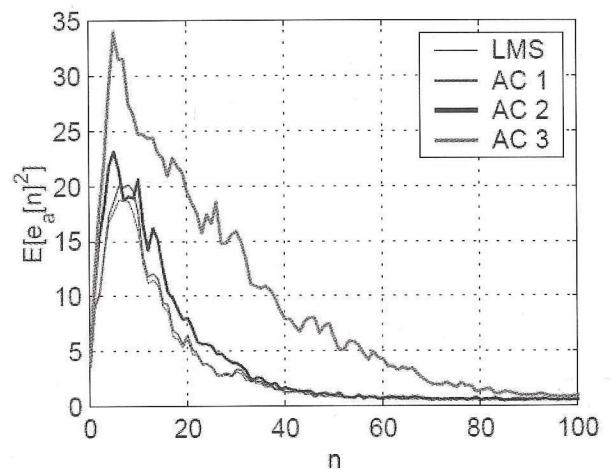


Fig. 2. Curvas de aprendizaje de los algoritmos AC y LMS con parámetros de la Tabla 1. Media de 200 experimentos.

$\alpha$	$M_a^{(1)}$	$M_a^{(2)}$	$M_{lms}^{(1)}$	$M_{lms}^{(2)}$	$\mu = \mu_a$
$\alpha_o/4$	0,040	0,021	0,051	0,027	0,004
$\alpha_o/2$	0,074	0,040	0,093	0,054	0,007
$\alpha_o$	0,129	0,074	0,161	0,101	0,013
$2\alpha_o$	0,208	0,130	0,251	0,208	0,020
$4\alpha_o$	0,297	0,210	0,344	4,429	0,027
$\sigma_x^2 = 4, \sigma_\eta^2 = 0,5, \gamma = 0,01, m_1 = 6, \alpha_o = 0,0313$					

Tabla 2. Desajuste de los algoritmos AC ( $M_a$ ) y LMS ( $M_{lms}$ ). Media sobre 200 experimentos. (1) Valor teórico. (2) Valor medido.

Este desvío se debe probablemente a las hipótesis simplificadoras utilizadas. Por ejemplo, a suposición de que  $x(n)$  sea blanco no corresponde a la realidad. Aquí un análisis más cuidadoso de los momentos de primera y segunda ordenes se hace necesario.

Por otro lado, se debe también observar que para pasos de adaptación  $\mu$  para los cuales el comportamiento del algoritmo LMS es ruín, el algoritmo AC es estable y converge. Un ejemplo es presentado en la Fig.3. Las curvas de aprendizaje mostradas fueron obtenidas en las siguientes condiciones:  $m_1 = 460$ ,  $\alpha = 0,3125$ ,  $\gamma = 0,1$ ,  $\mu = 0,0416$ ,  $\sigma_x^2 = 4$  y  $\sigma_{\eta}^2 = 0,5$ . El condicionamiento de la matriz de correlación  $R_x$  es dado por  $\lambda_{\max}/\lambda_{\min} = 6,6$ . El desajuste obtenido para el algoritmo AC fue  $M_a = 0,64$ , en cuanto que para el algoritmo LMS se obtuvo  $M_{lms} = 3,9 \cdot 10^4$ .

Dados estos resultados se puede ahora establecer un procedimiento para el ajuste de los parámetros del algoritmo AC. Se supone una velocidad de convergencia deseada correspondiente a un paso de adaptación  $\mu$  del algoritmo LMS y que se conozca la matriz de covariancia  $R_x$  de la señal de entrada o por lo menos

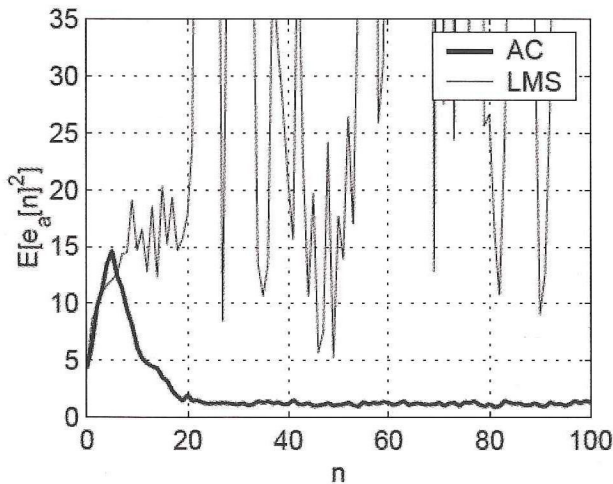


Fig. 3. Curvas de aprendizaje de los algoritmos AC y LMS con  $\mu = \mu_a$ . Media sobre 200 experimentos.

su variancia  $\sigma_x^2$  y una estimativa de  $\lambda_{\min}$ , el autovalor mínimo de  $R_x$ . En esta situación un procedimiento simple para la obtención de los parámetros del

algoritmo AC es el siguiente:

- Se toma  $\mu_a = \mu$  y se determinase el producto  $\alpha \gamma m_1$  utilizándose por ejemplo que  $\text{tr}\{R_x\} = M\sigma_x^2$ .
- Se determina  $\alpha$ ,  $\gamma$  y  $m_1$  satisfaciendo la condición (32) con una cierta folga para garantizar que la velocidad de convergencia será la deseada y procurando obtener un valor bajo para el desajuste.

Como muestra el ejemplo de la Figura 3 el valor de  $\mu$  puede ser tal que el algoritmo LMS sea en principio rápido mas no presente un buen comportamiento para la señal de entrada en cuestión. En esta situación es que el algoritmo AC presentará las mayores ventajas sobre el algoritmo LMS.

### Conclusión

Fue mostrado que el algoritmo acelerador es un algoritmo del tipo cuasi-Newton presentando-se as estimativas de gradiente y matriz Hessiana que utiliza. Además, fue introducida una versión simplificada del algoritmo AC que es más simple de ajustar que su forma general. Por medio de un análisis de momentos de primera y segunda ordenes fueron obtenidas expresiones para el error cuadrático medio y para el desajuste. Estos resultados fueron confirmados cualitativamente por los resultados de simulaciones habiendo un cierto desvío cuantitativo. Esta discrepancia indica que algunas hipótesis simplificadoras hechas en el análisis de los momentos deben ser abandonadas en estudios futuros. De cualquier forma, los resultados obtenidos generan indicaciones esclarecedoras que facilitan el ajuste de los parámetros del algoritmo AC. Además, el comportamiento robusto de convergencia del algoritmo propuesto hace de él una buena alternativa para aplicaciones donde las señales de entrada son mal condicionadas y el algoritmo LMS presenta problemas de convergencia.

## 6. Bibliografía

- GERKEN, M., PAIT, F.M. y JOJOA GOMEZ, P.E. (April 2000). An Adaptive Algorithm with Parameter Acceleration, Proceedings of ICASSP'2000, Vol I, pp. 17-20
- PAIT, F.M. (1998). A Tuner that Accelerates Parameters, Systems & Control Letters 35, pp. 65-68.
- LUENBERGER, D.G. (1989). Linear and Nonlinear Programming, 2. Ed. Addison-Wesley.
- SOLO, V. y KONG X. (1995). Adaptive Signal Processing Algorithms, Stability and Performance, Prentice Hall.
- SAMSON, C. y REDDY, V. U. (October 1983). Fixed point error analysis of the normalized ladder algorithms, IEEE Trans. on ASSP, vol. 31, n. 5, October 1983, pp. 1177-1191.
- HAYKIN, S. (1996) Adaptive Filter Theory, 3. Ed., Prentice Hall.