

Procesamiento de Señales II
TP2: Filtrado Adaptativo

Sampayo, Sebastián Lucas

Segundo Cuatrimestre 2015



**FACULTAD
DE INGENIERIA**

Universidad de Buenos Aires

Índice

1. Algoritmo LMS	1
1.1. Mismatch	1
1.2. Límite de estabilidad	2
1.3. Misadjustment	2
2. Algoritmo NLMS y APA	4
2.1. LMS vs NLMS	4
2.2. APA	5
3. Algoritmo RLS	8
4. Ejemplo de aplicación: cancelación de una senoide	13
4.1. Planteo del problema	13
4.2. Ajuste de parámetros	13
4.3. Comportamiento frente a un cambio en la entrada	15
4.4. Notch	16
5. Conclusiones	20

1. Algoritmo LMS

1.1. Mismatch

Para estudiar este algoritmo se simularon una serie de ejercicios a través del programa MATLAB. En primera instancia se tomaron 4 valores de μ pequeños y se midió el *mismatch* promedio en dB por medio de una simulación Monte-Carlo, esto es, repitiendo el algoritmo y promediando. En particular, se tomaron 100 repeticiones del algoritmo para cada μ , con 20000 iteraciones del filtro en cada caso. Los parámetros de entrada para el experimento fueron:

- Proceso AR1 de entrada, con coeficiente $a = 0,3$, varianza de ruido de unitaria.
- SNR = 20dB.
- Filtro óptimo provisto como dato.

Los resultados del experimento se muestran a continuación:

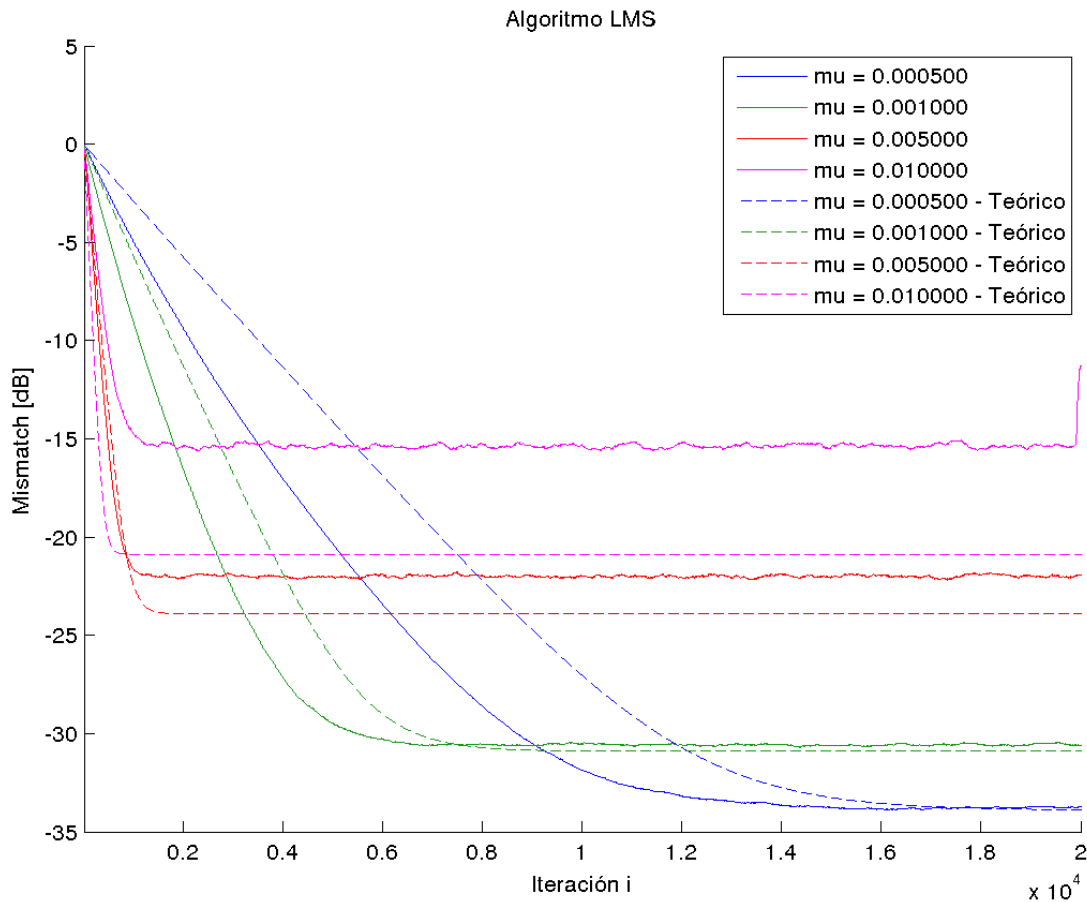


Figura 1: Mismatch promedio para el LMS.

En la figura puede apreciarse también las curvas teóricas que corresponden a dichos valores de μ a modo comparativo. En este sentido podemos analizar 3 aspectos del resultado: el error en estado estable (con i tendiendo a infinito), la velocidad de convergencia y la correspondencia con el modelo teórico.

Respecto al error obtenido, la simulación comprueba que a medida que aumenta el tamaño del paso μ , este error crece (esto se puede ver en las últimas iteraciones).

En cuanto a la velocidad de convergencia, la teoría indica que esta magnitud debería aumentar al aumentar μ , lo cual se evidencia para los 3 valores más chicos. No obstante, para $\mu = 0,01$ se obtiene una velocidad de

convergencia igual que para $\mu = 0,005$, con el agregado de un error final mucho mayor y más alejado del valor teórico. Esto puede atribuirse a que el valor del parámetro supera el nivel admisible por el modelo teórico (*small step-size theory*), el cual se basa en valores pequeños de μ .

En conclusión podemos decir que la validez de la teoría para estos parámetros de entrada y SNR, se mantiene hasta valores de $\mu \approx 0,005$.

1.2. Límite de estabilidad

El modelo de la *small step-size theory* indica que el límite de estabilidad para el algoritmo se alcanza con

$$\mu_{lim} = \frac{2}{\lambda_{max}}$$

que en nuestro caso toma el valor de 0.98. Este valor en verdad no nos aporta demasiado, dado que en la práctica vemos que el tamaño de paso para el cual el filtro se torna inestable es del orden del 1.3% de este valor, como se puede ver en la siguiente simulación, utilizando el mismo proceso de entrada que antes, pero reduciendo la cantidad de repeticiones e iteraciones:

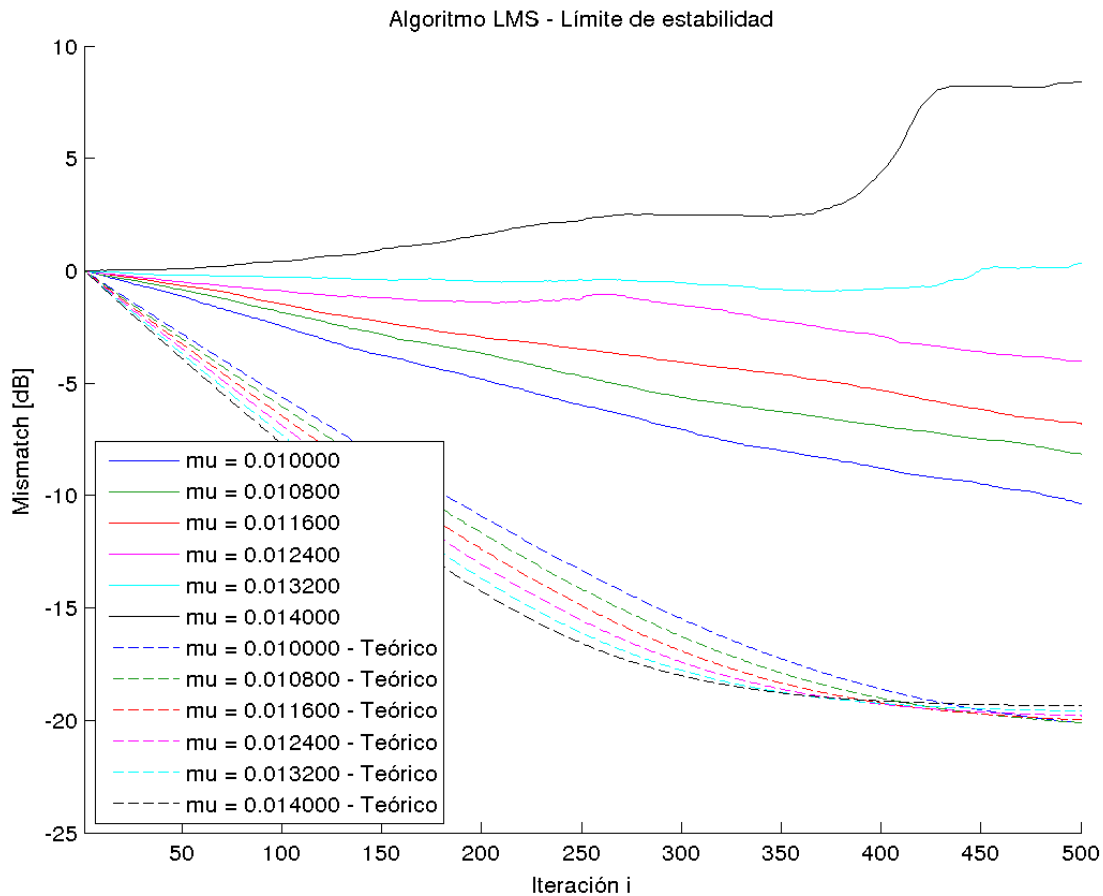


Figura 2: Límite de estabilidad para el LMS.

1.3. Misadjustment

Con el objetivo de seguir comparando los resultados empíricos con el modelo teórico, se estudió la dependencia del *misadjustment* con el tamaño del paso recursivo μ . Según la teoría, esta relación está dada por:

$$\mathcal{M} = \frac{\mu}{2} \text{tr}(R_u)$$

Mediante simulaciones Monte Carlo, se tomaron los valores finales del error de las últimas iteraciones en cada μ , se las promedió y con esto se construyó la siguiente gráfica:

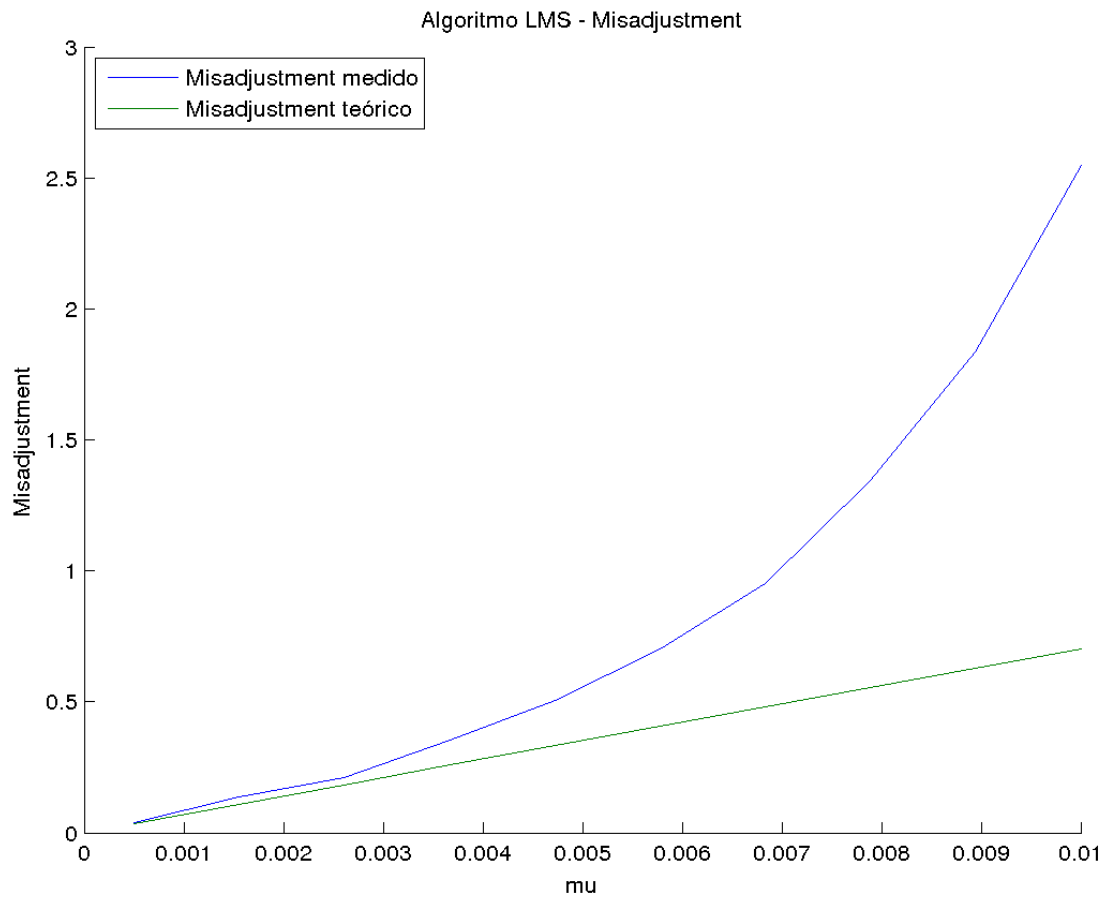


Figura 3: Dependencia del Misadjustment con μ .

Como se puede ver, la práctica nos indica que esta dependencia es más bien cuadrática que lineal.

2. Algoritmo NLMS y APA

2.1. LMS vs NLMS

En esta simulación se aumentó el “color” del proceso de entrada, llevando al coeficiente del AR1 a $a = 0,9$, y se ajustó el valor de μ de modo que el algoritmo LMS sea estable en media cuadrática. Para esto se graficaron las curvas del mismatch promedio mediante simulaciones Monte-Carlo con distintos valores de tamaño de paso, hasta encontrar el límite de estabilidad.

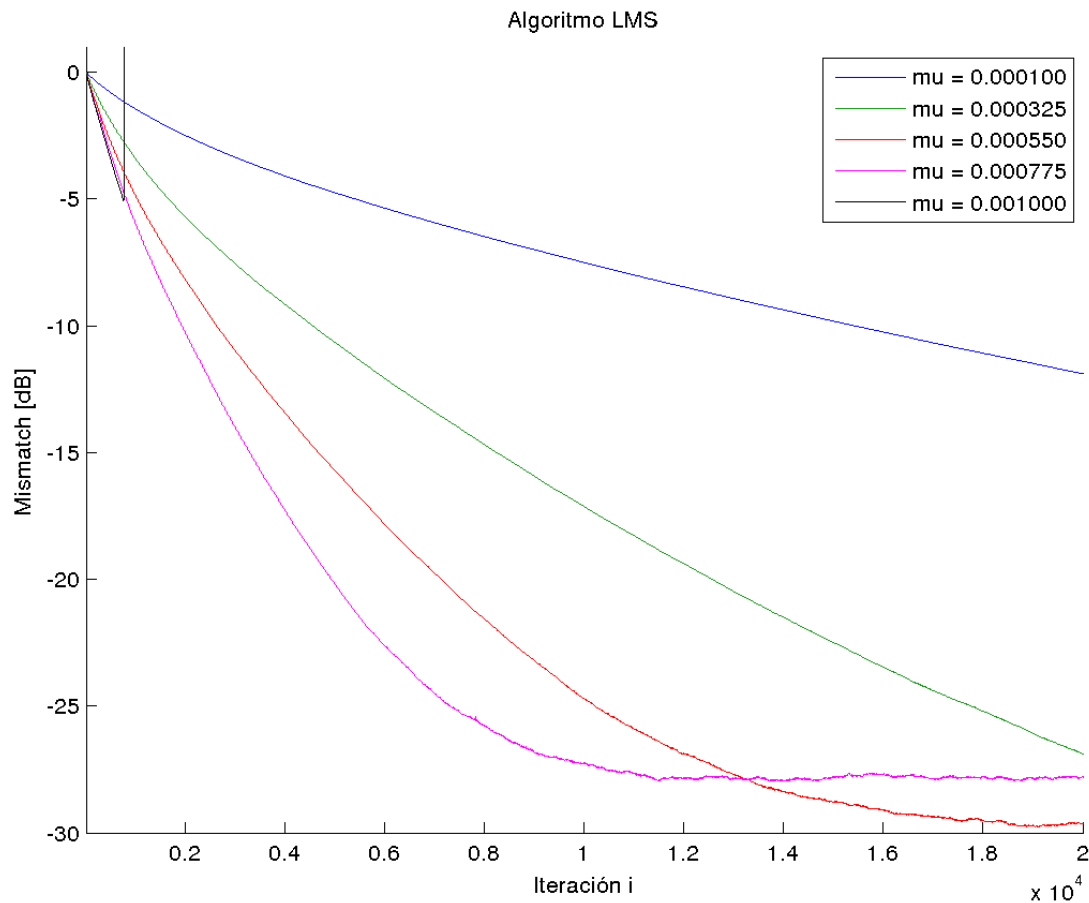


Figura 4: Límite de estabilidad para el LMS con AR1(0.9).

Luego se simuló nuevamente el mismatch promedio, pero utilizando el algoritmo NLMS para μ : 0.1 y 1, obteniendo el siguiente gráfico:

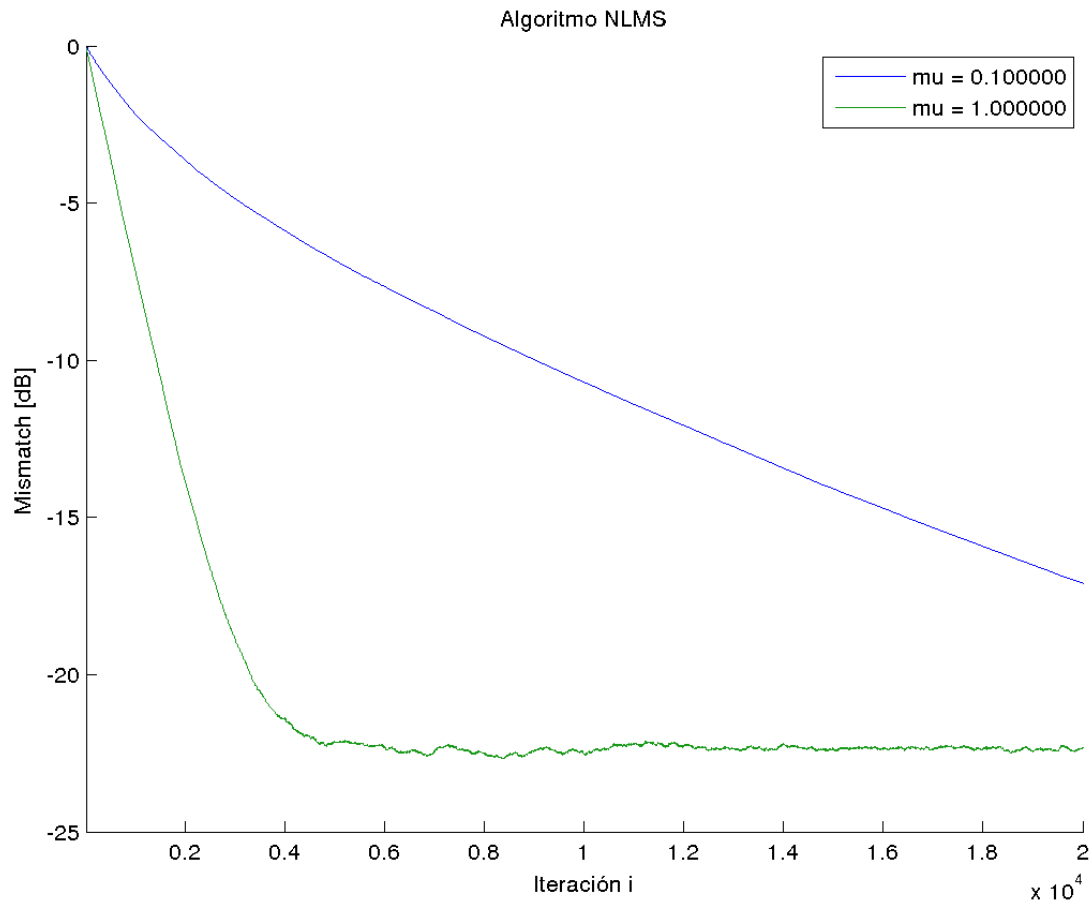


Figura 5: Mismatch promedio para el NLMS.

Como se puede ver, el NLMS no diverge en ningún caso, por lo tanto desde el punto de vista de la estabilidad es altamente superior al LMS, el cual a partir de $\mu = 0,001$ diverge.

Por otro lado, la velocidad de convergencia es mayor para el NLMS ya que podemos ver que como máximo este algoritmo alcanza la estabilidad en la iteración 4000 mientras que el caso más rápido del LMS lo hace para la iteración 10000.

2.2. APA

Considerando ahora el algoritmo APA para orden 1, 2 y 4, y con un proceso de entrada aún más coloreado, AR1(0.95), se graficó nuevamente el mismatch promedio para cada caso con $\mu = 0,1,0,5,1$. A continuación se muestran las curvas obtenidas:

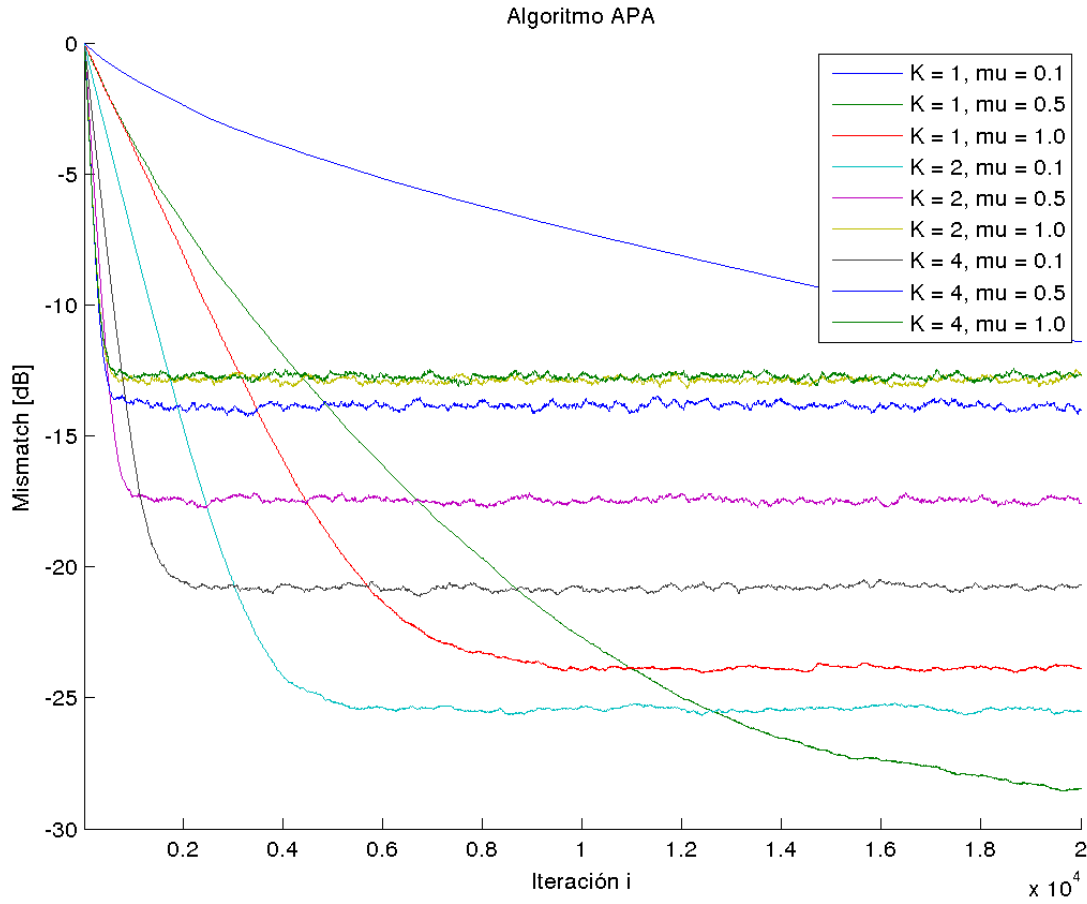


Figura 6: Mismatch promedio para el APA.

Podemos concluir a partir del resultado que la velocidad de convergencia del algoritmo aumenta a medida que crece el orden del APA. Sin embargo, esta mejoría se reduce a medida que consideramos ordenes más altos.

Respecto a la coloración de la entrada, podemos ver que este algoritmo se adapta mucho mejor, obteniendo velocidades de convergencia mayores que el LMS a costa de errores finales ligeramente mayores. En el gráfico anterior del LMS podíamos ver que lo mejor que puede hacer este algoritmo, para una entrada de coloración similar, es converger en la iteración 10000 aproximadamente, con un mismatch de -27dB. Esta curva la podríamos comparar con la curva del APA de $K=2$ y $\mu=0.1$, cuyo mismatch es similar y alcanza la convergencia en la iteración 5000, es decir el doble de rápido que el LMS. En el último gráfico también podemos ver la capacidad del APA de aumentar aún más su velocidad, pero a costa de aumentar el mismatch final. Estas velocidades solo son alcanzables por el LMS cuando la entrada es más blanca, como se vio en la primera sección de este trabajo.

Es importante destacar, que el aumento de estas prestaciones que ofrece el APA trae aparejado un costo computacional $\mathcal{O}(M \cdot K^2)$ mucho mayor al del LMS que es $\mathcal{O}(M)$

Por otro lado, se estudió la dependencia del valor asintótico del mismatch promedio con el tamaño del paso μ . Para esto nuevamente se realizaron simulaciones Monte-Carlo para valores de $K=1, 2, 4$ y 8 , y barriendo el μ entre 0 y 2. A continuación se muestra el resultado:

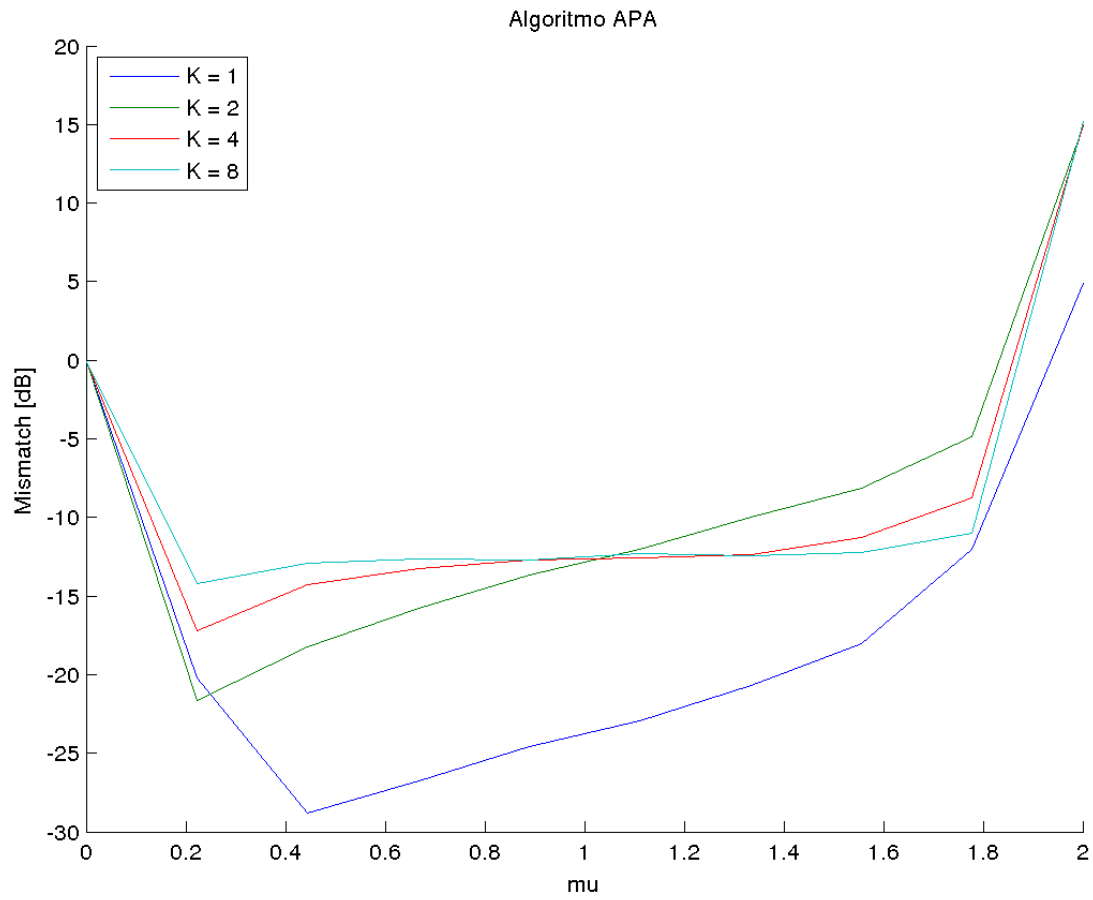


Figura 7: Mismatch promedio asintótico para el APA en función de μ .

En la figura podemos comprobar un aumento del valor asintótico del mismatch promedio con el aumento de μ entre 0.5 y 2. Sin embargo, también podemos apreciar que con el aumento del orden del APA, esta curva tiende a aplanarse cerca de un mismo valor de mismatch, de aproximadamente -15dB.

3. Algoritmo RLS

En esta sección se comparó el mismatch promedio entre los distintos algoritmos LMS, RLS y APA de orden 1, 2 y 4. En cada caso se optimizaron los parámetros correspondientes a cada uno para alcanzar la máxima velocidad inicial de convergencia. A continuación se muestra el resultado para una SNR de 30dB y entrada blanca:

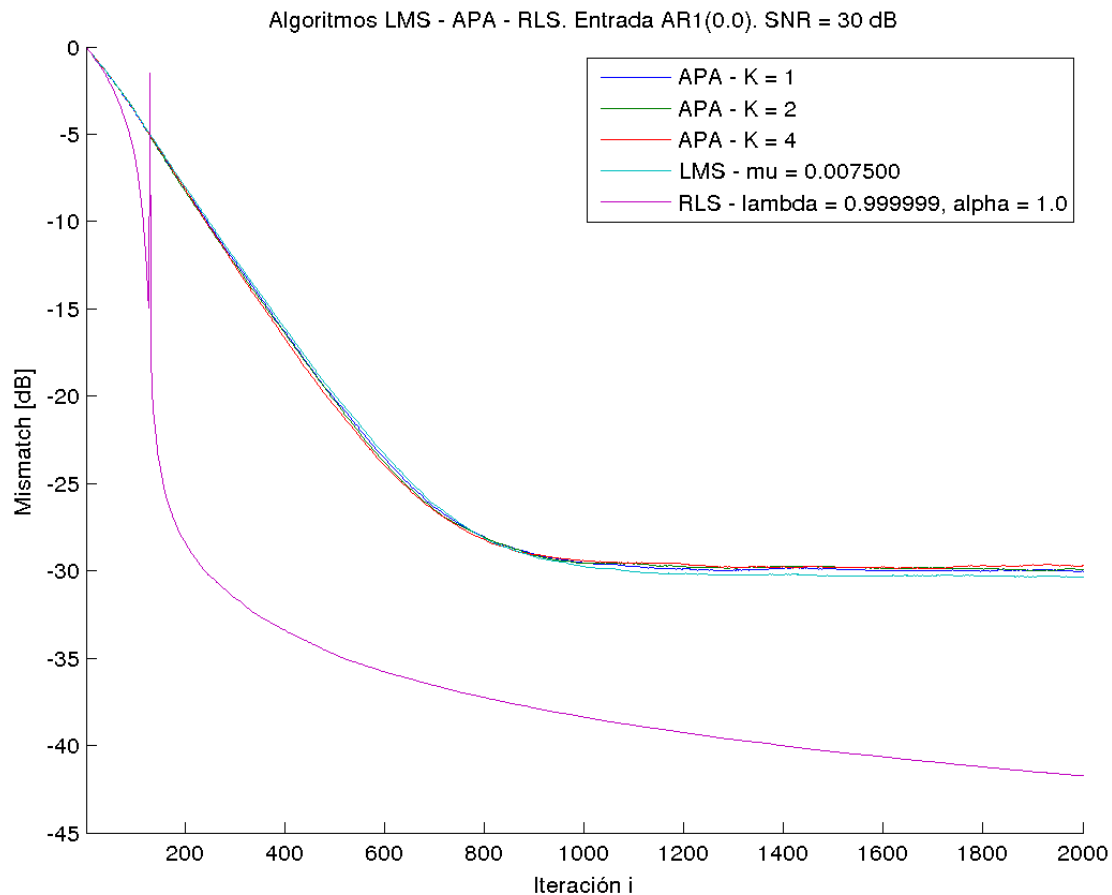


Figura 8: Mismatch promedio para entrada blanca y SNR de 30dB.

En la imagen anterior se puede ver que para entrada completamente descorrelacionada, los algoritmos APA y LMS no hacen diferencia alguna, mientras que el RLS les saca una gran ventaja tanto en velocidad de convergencia (casi de 10 veces mejor) como en error final.

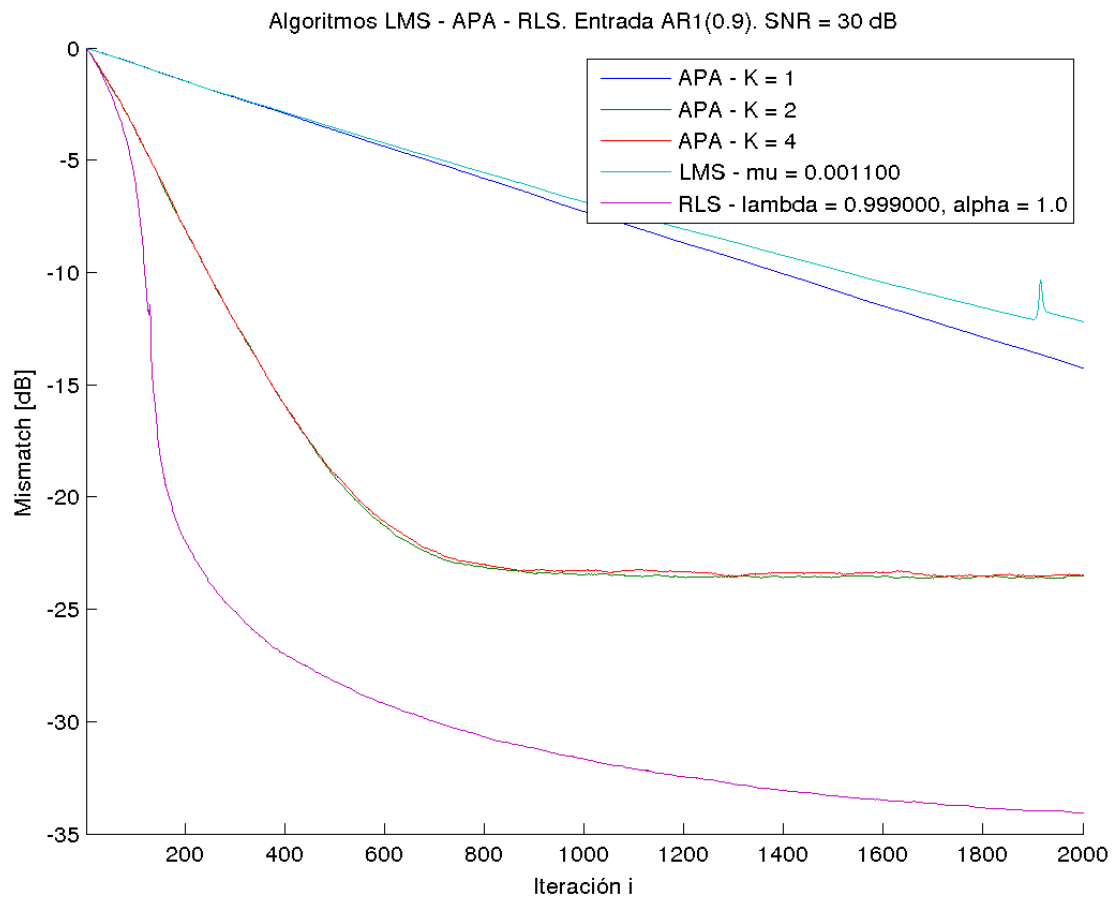


Figura 9: Mismatch promedio para entrada coloreada y SNR de 30dB.

Para el caso de entrada coloreada, el LMS y el APA de orden 1 no consiguen acercarse a la performance que logran el resto de los algoritmos. El RLS y el APA de orden 2 y 4 mantienen el mismo resultado que en el caso de entrada blanca, con lo cual podemos decir que son más robustos respecto a la coloración de la entrada.

Por otro lado, al aumentar el nivel de ruido, con una SNR de 5dB, se espera que empeore el desempeño en todos los casos, y efectivamente esto es lo que sucede. En primer lugar se realizó la simulación con una entrada blanca, y se obtuvo lo siguiente:

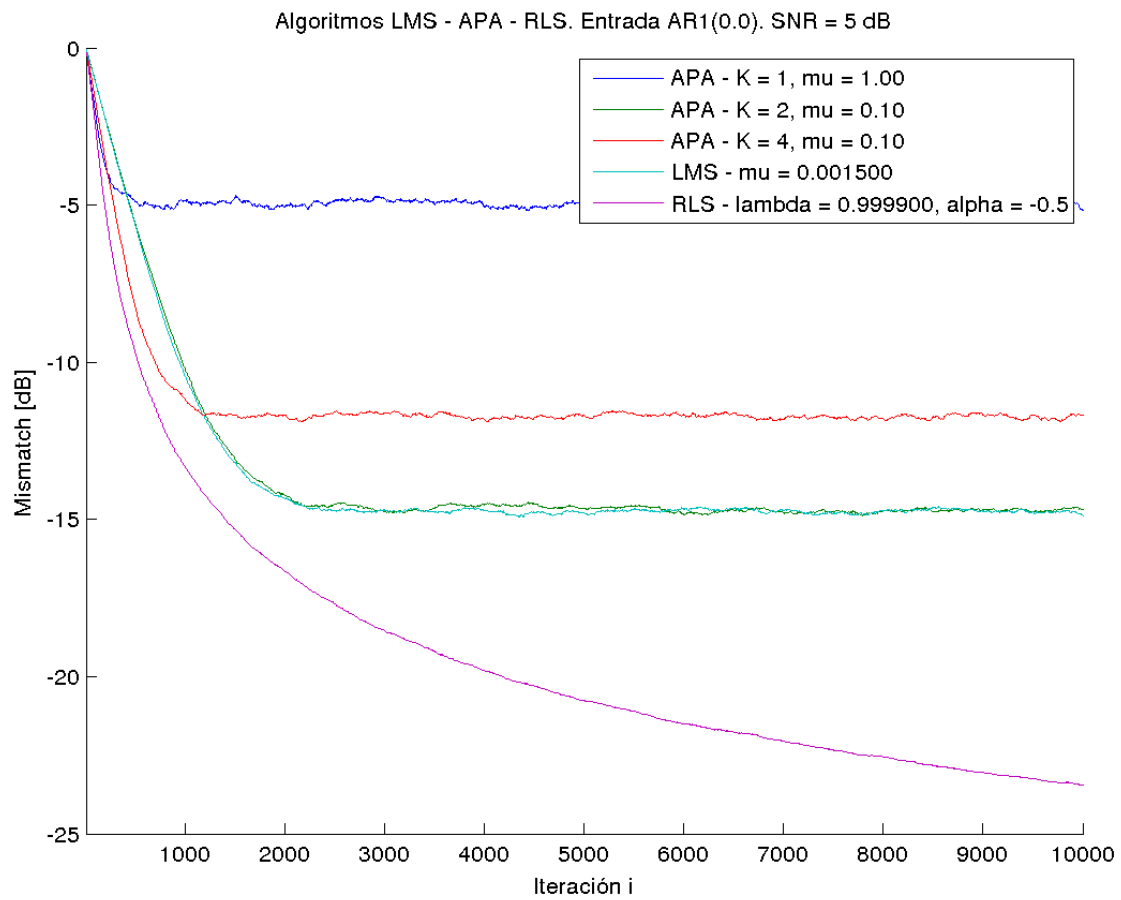


Figura 10: Mismatch promedio para entrada blanca y SNR de 5dB.

Esencialmente podemos apreciar 2 grandes cambios respecto a las simulaciones anteriores: el mismatch y la velocidad de convergencia empeoran en todos los casos. Por ejemplo, en el APA, que en el caso de entrada blanca, SNR=30dB, convergía en todos los ordenes alrededor de la iteración 1000 con un mismatch de -30dB, ahora solo alcanza esa velocidad cuando su orden es 4 y el mismatch resultante es de -12dB, o con orden 1 y mismatch de -5dB.

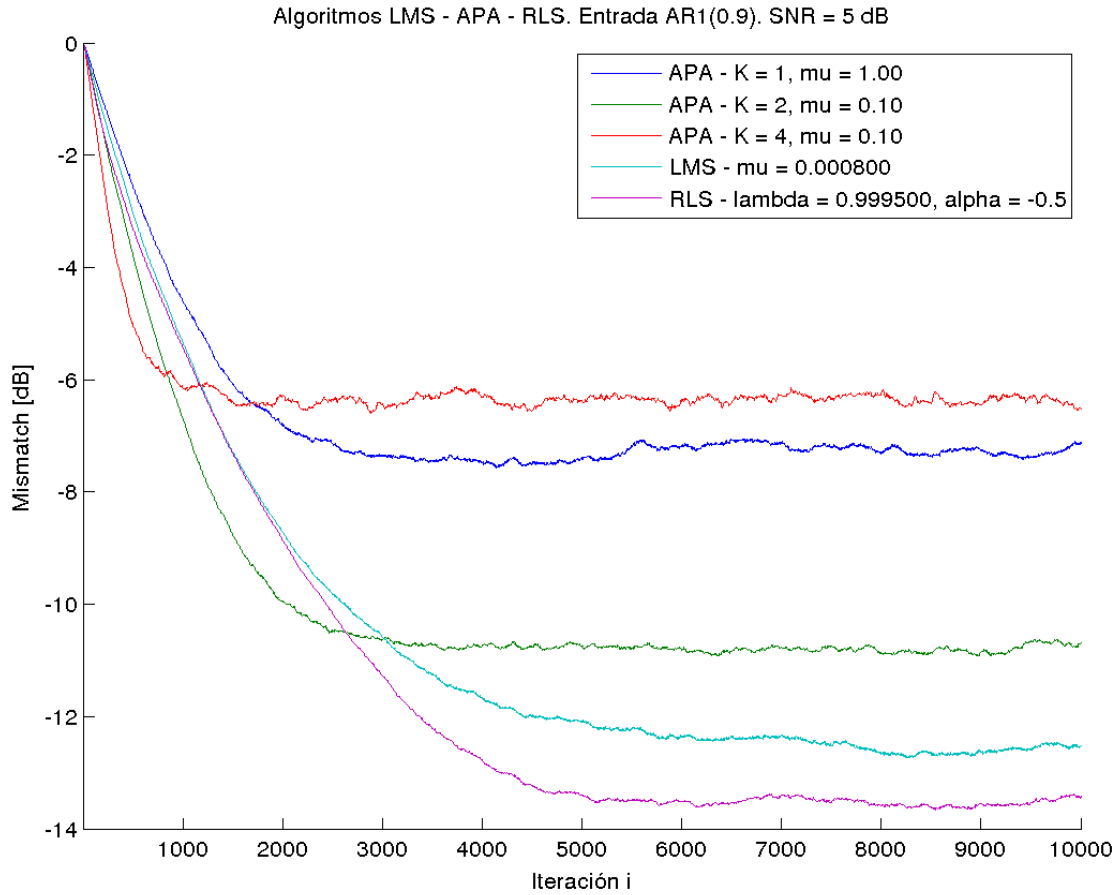


Figura 11: Mismatch promedio para entrada coloreada y SNR de 5dB.

Finalmente en el peor de los escenarios posibles, con entrada coloreada y relación señal ruido más chica, el algoritmo más rápido fue el APA de orden 4, y el de menor error final fue el RLS.

Para concluir, se realizó la siguiente tabla con un resumen de los resultados obtenidos en toda la sección. En líneas generales, podemos afirmar que el RLS es el algoritmo con mayor velocidad de convergencia (excepto en el peor escenario posible) y menor error final, pero también, aquel de mayor coste computacional.

Algoritmo	Iteración en la que converge				Mismatch final				Orden de Convergencia
	SNR = 30dB		SNR = 5dB		SNR = 30dB		SNR = 5dB		
	a = 0	a = 0.9	a = 0	a = 0.9	a = 0	a = 0.9	a = 0	a = 0.9	
LMS	1000	10000	2000	5000	-30	-20	-15	-12	$\mathcal{O}(M)$
APA 1	1000	10000	500	1500	-30	-20	-5	-7	$\mathcal{O}(M)$
APA 2	1000	700	2000	2500	-30	-23	-15	-11	$\mathcal{O}(4M)$
APA 4	1000	700	1000	1000	-30	-23	-12	-6	$\mathcal{O}(16M)$
RLS	400	400	500	5000	-40	-35	-25	-14	$\mathcal{O}(M^2)$

Cuadro 1: Resumen de resultados comparativos.

Nota: Cabe destacar que el valor de la “iteración de convergencia” en los casos del RLS, son orientativas, dado que el algoritmo tarda mucho en converger estrictamente. Sin embargo, al ver la pendiente de la recta inicial del mismatch, se observa que la velocidad inicial con la que reduce su error es muy alta, y por lo tanto el

valor para poner en la tabla se ha elegido en comparación con las velocidades iniciales de los otros algoritmos. Además, la intención es comparar velocidades iniciales.

4. Ejemplo de aplicación: cancelación de una senoide

4.1. Planteo del problema

Para este problema se consideró el siguiente diagrama en bloques:

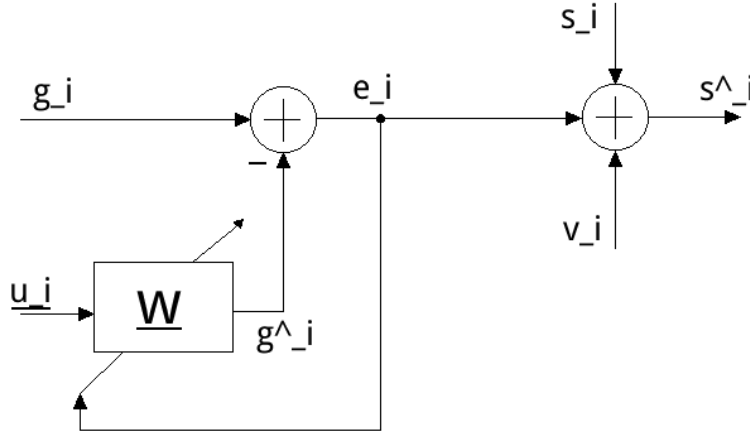


Figura 12: Diagrama en bloques del problema de estimación.

que lleva a las siguientes relaciones:

w_0 : Filtro óptimo.

$$d_i = s_i + g_i + v_i$$

$$g_i = \underline{w}_0^H \underline{U}_i$$

donde

$$\underline{U}_i = [u_i u_{i-1} \cdots u_{i-M+1}]^T$$

y u_i es la entrada de referencia, correlacionada con g_i . Por lo tanto, los estimadores estarán dados por:

$$\hat{g}_i = \underline{w}_i^H \underline{U}_i$$

$$\hat{s}_i = d_i - \hat{g}_i$$

Y en conclusión, la señal error es:

$$e_i = g_i - \hat{g}_i$$

De esta forma, una hipótesis que deberíamos tener en cuenta es que s_i y g_i estén descorrelacionadas.

4.2. Ajuste de parámetros

Mediante un barrido de distintas longitudes de filtro y distintos valores del tamaño del paso del LMS μ , se midió la distancia normalizada entre la señal de ECG original y la estimada, esto es:

$$Distancia_i = \frac{|\underline{s}_i - \hat{\underline{s}}_i|^2}{|\underline{s}_i|^2}$$

$$\underline{s}_i = [s_i s_{i-1} \cdots s_0]^T$$

a lo que se le aplicó 10 veces el logaritmo en base 10 para visualizarlo mejor y se graficaron las siguientes curvas, utilizando siempre una amplitud A de la señal de interferencia de A=1:

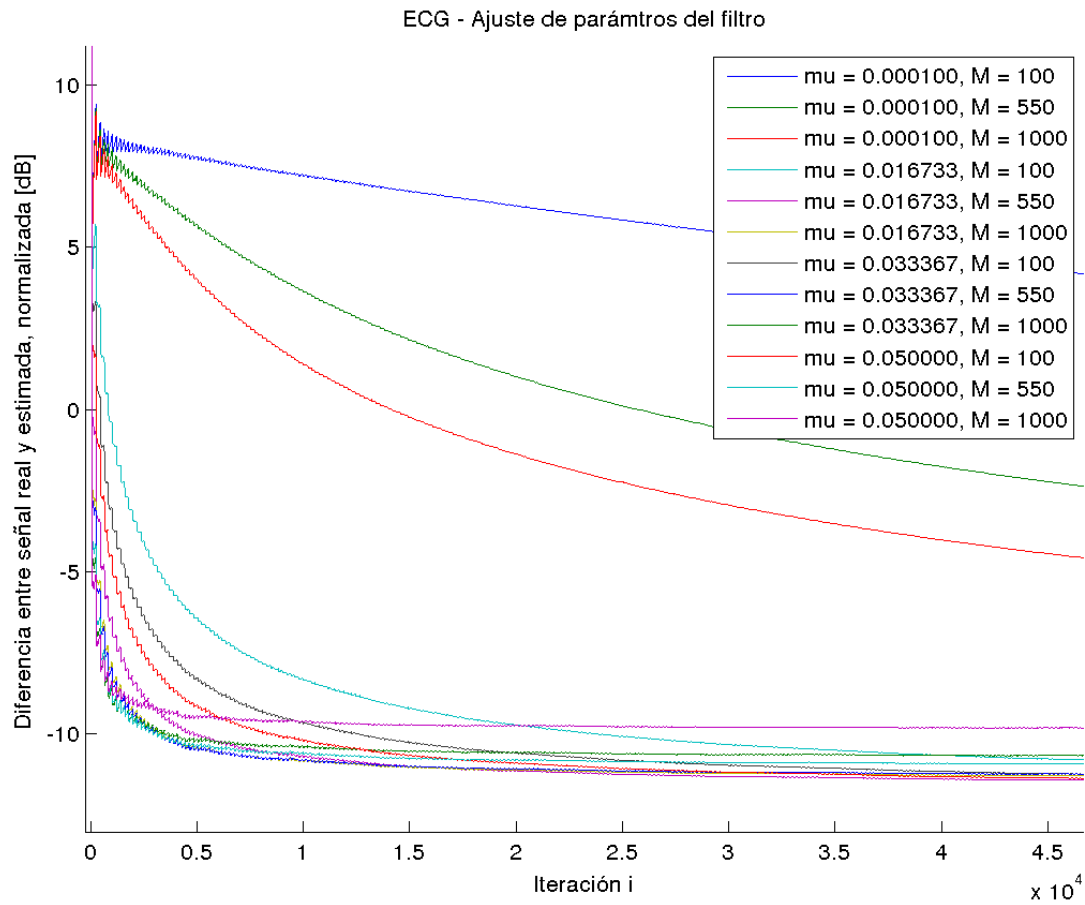


Figura 13: Ajuste de parámetros del filtro.

Es importante destacar que la variación en la fase de la señal de interferencia no modificó sustancialmente los resultados.

A partir de este experimento se eligió el siguiente juego de parámetros:

- $M = 500$
- $\mu = 0,05$

que dan el siguiente resultado:

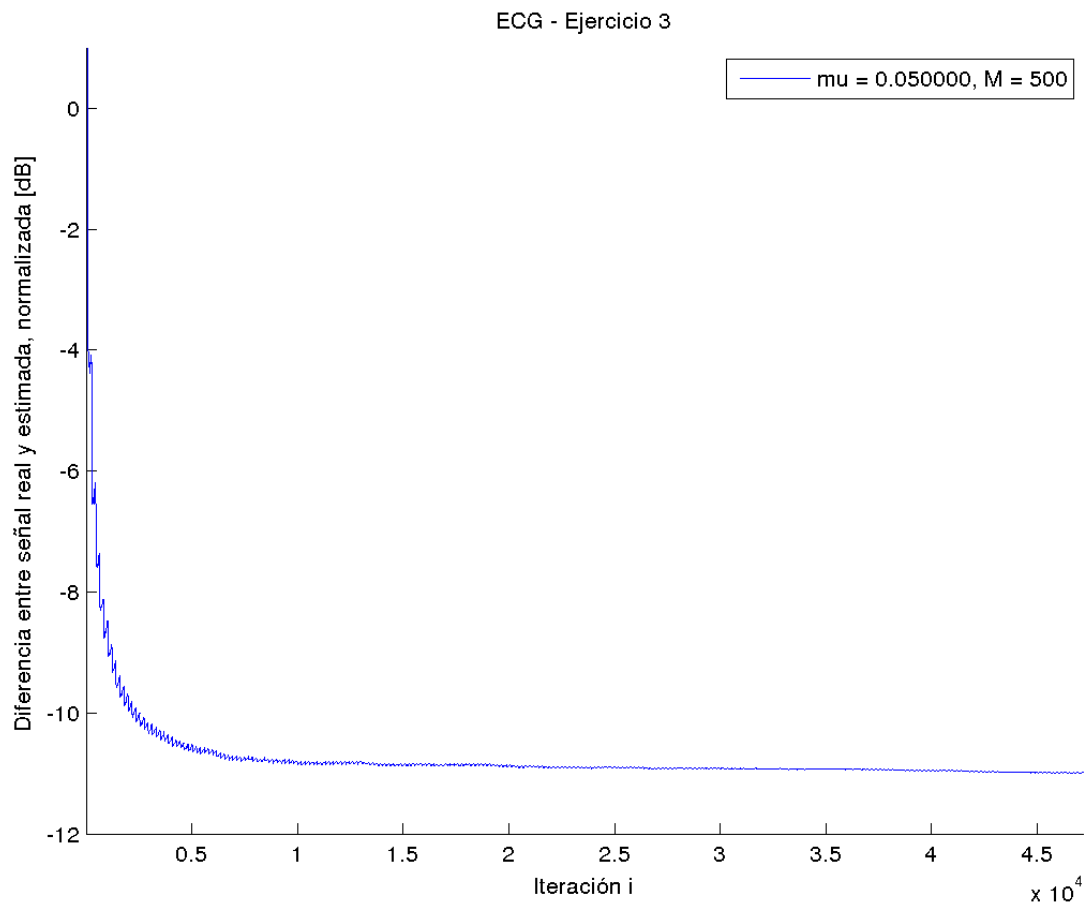


Figura 14: Error en la señal estimada en cada instante.

4.3. Comportamiento frente a un cambio en la entrada

Luego se realizó una corrida donde a la mitad de la simulación la frecuencia de la interferencia y la referencia cambian en forma abrupta a 70Hz. En este caso se puede observar un ligero aumento abrupto del error en dicho momento, y a partir de entonces progresivamente, el error vuelve a reducirse a su mínimo valor. En definitiva, el filtro se “adapta” a la señal.

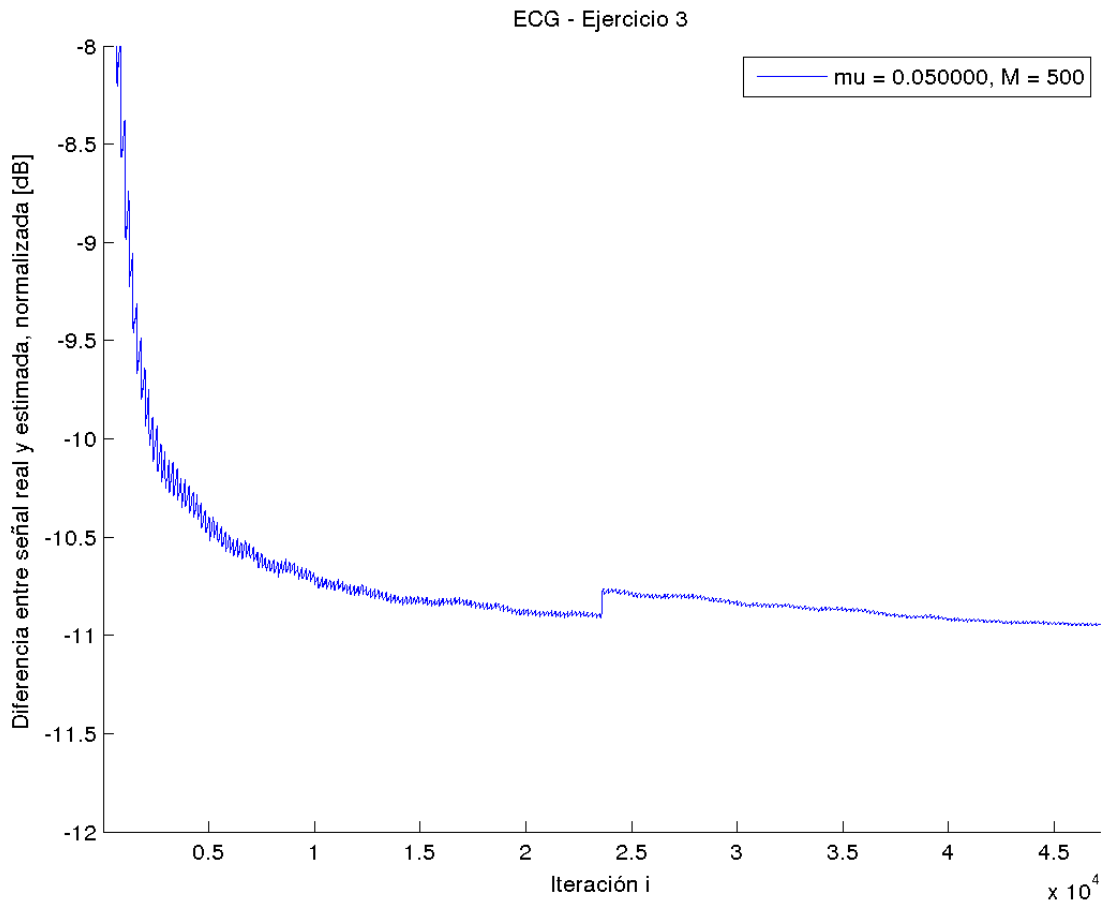


Figura 15: Cambio abrupto en la frecuencia de la interferencia

También se simuló el caso ideal en el que no hay ruido, es decir solo se tiene la señal del ECG original sumada a la interferencia, donde encontramos que el filtro se comporta aún mejor, alcanzando un error de -20 dB con una velocidad de convergencia igual.

4.4. Notch

Por otro lado se implementó un filtro notch digital a la frecuencia nominal de 60 Hz, de forma tal que solo elimine esa frecuencia y no altere la forma de la señal original de ECG. Para esto se utilizó el algoritmo de Parks-McClellan, buscando que en la banda de 60 Hz la respuesta sea 0, con un ancho de banda de 2 Hz. Para esto se eligió el mismo orden de filtro que el utilizado para el algoritmo LMS, es decir, $M = 500$. De este modo se filtró en cada instante, las últimas M muestras de la señal recibida (d), con el filtro notch generado y se calculó el error con la misma medida que antes. En primer lugar, se muestra la respuesta en frecuencia del filtro notch implementado junto con la especificación ideal propuesta. A continuación se muestran los resultados para todos los escenarios propuestos (con y sin ruido, con y sin cambio abrupto de frecuencia).

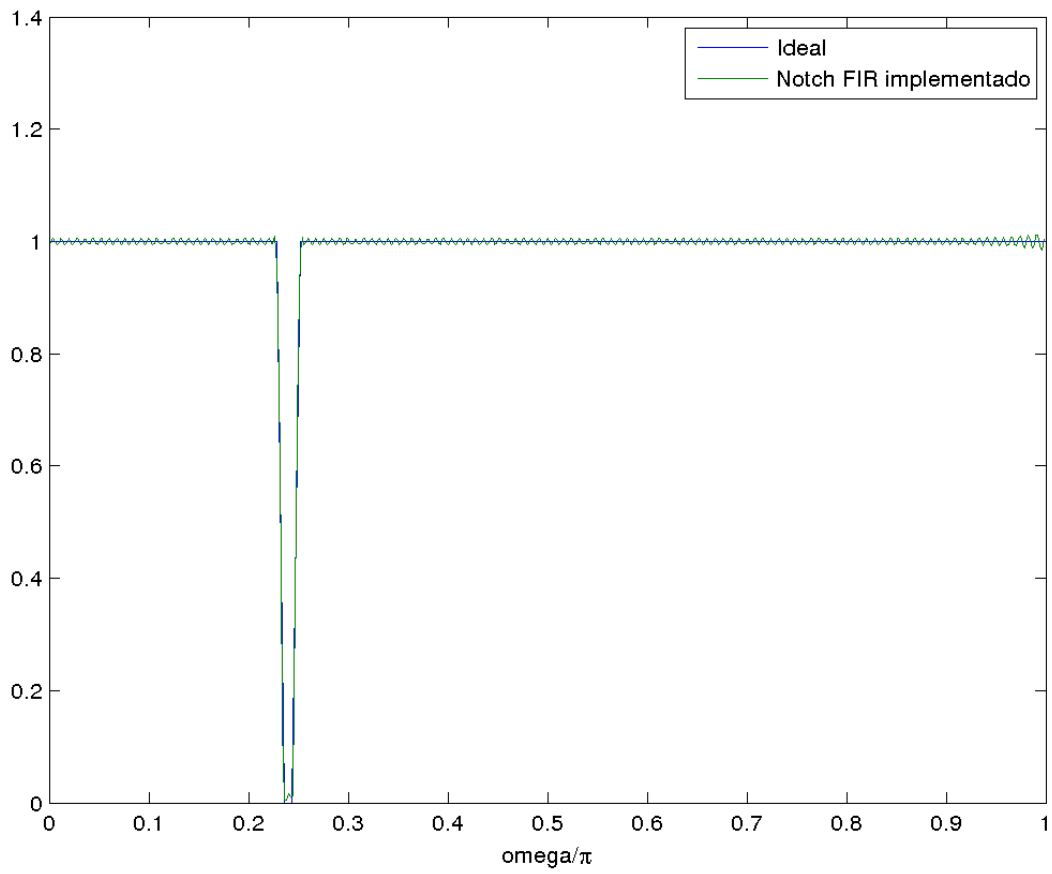


Figura 16: Filtro digital Notch, FIR, para una frecuencia de muestreo de 500 Hz.

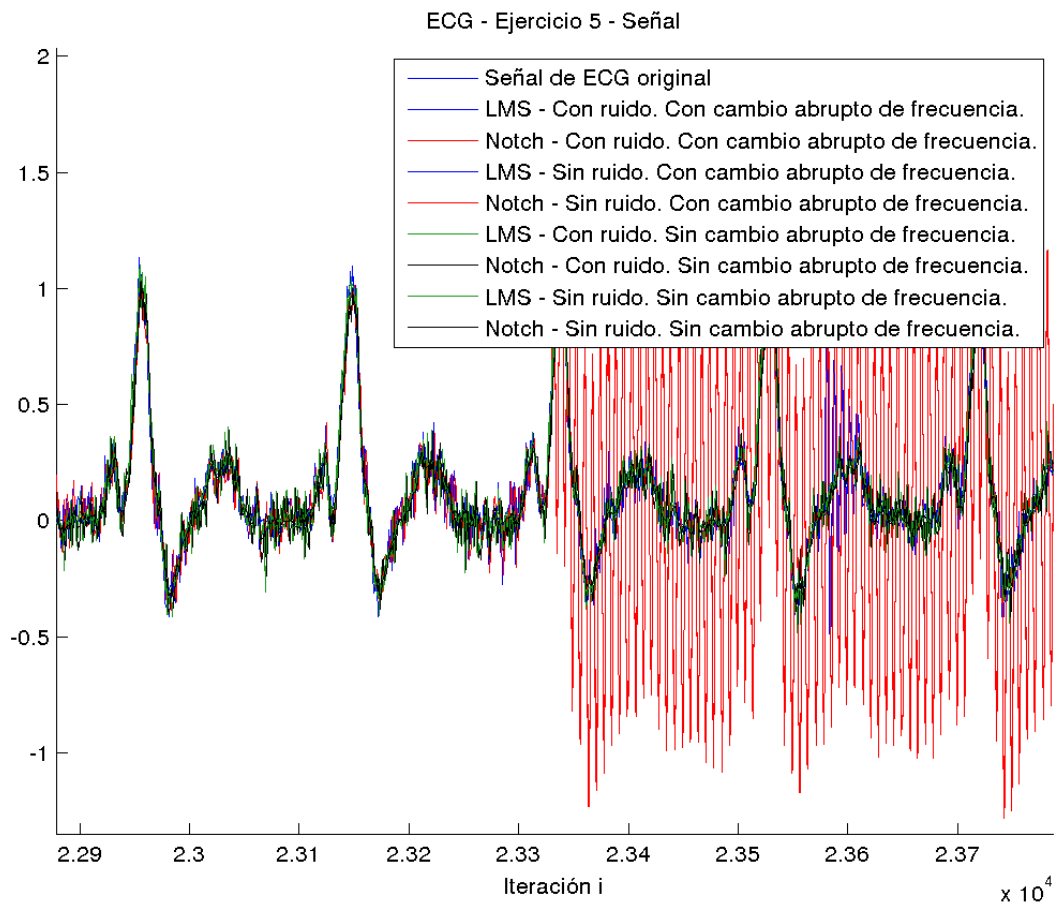


Figura 17: Señales obtenidas.

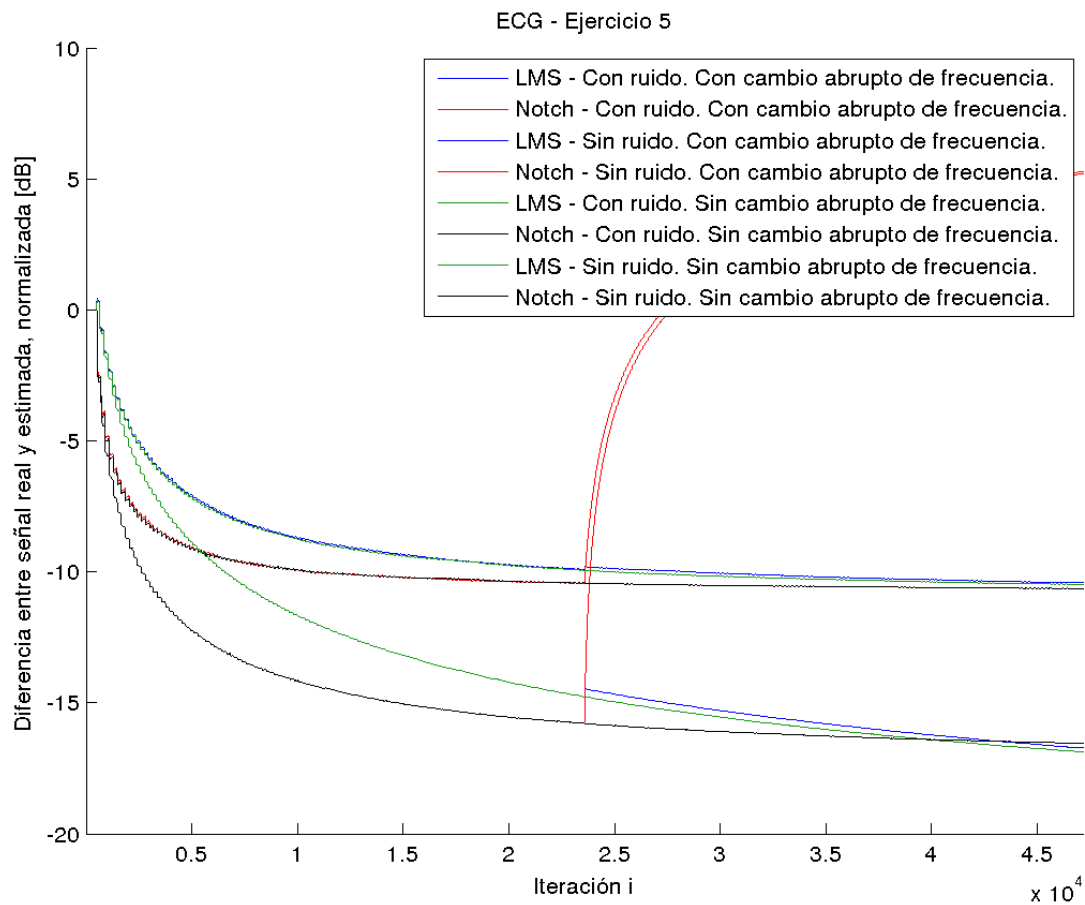


Figura 18: Error en la señal estimada en cada instante.

En las figuras puede verse que cuando la frecuencia de la interferencia se mantiene en 60 Hz:

- El Notch tiene una velocidad de convergencia mayor que el LMS
- Ambos tienen el mismo error final en estado estable

Sin embargo, cuando se presenta un cambio en la frecuencia de la señal, el notch queda completamente inutilizable, y por el contrario, el LMS se adapta perfectamente al cambio.

5. Conclusiones

En conclusión, todos los filtros estudiados presentan características distintas, algunos mejores que otros, con relaciones de compromiso entre velocidad de convergencia, error en estado estable y orden de complejidad computacional.

En cuanto a la elección de un algoritmo, esta dependerá específicamente de los requisitos de la aplicación. Por ejemplo si no hubiese que preocuparse por el recurso computacional, definitivamente el RLS siempre sería la mejor opción, con valores de sus parámetros ajustados correctamente. Sin embargo, en caso de que este coste sea una restricción, podríamos considerar algún orden del APA que no sea tan complejo pero satisfaga requerimientos de velocidad y error mejor que el LMS.

Finalmente, en comparación con el filtro digital notch utilizado en la aplicación de ECG, podríamos concluir que en el caso particular del filtrado de la interferencia con la red eléctrica la cual puede cambiar su frecuencia inesperadamente, la mejor elección sería el LMS, el cual se adapta perfectamente a estos imprevistos. Igualmente si la aplicación fuese distinta, y se conociese con exactitud la frecuencia de la interferencia y que no varía, entonces el filtro notch sería seguramente una mejor elección debido a su mayor rapidez.