

Identificación de la frecuencia de corte de la pared arterial mediante filtrado adaptivo no recursivo con LMS

M. Federico Baguear *, Manuel Alfonso*, Damian Craiem*
Universidad Favaloro, Solís 453, (1078) Buenos Aires, Argentina, bagear@gmail.com

Resumen—Se realizó un algoritmo de identificación de sistemas mediante un filtro adaptivo no recursivo el cual ajusta sus coeficientes con el método de mínimos cuadrados. El objetivo de este trabajo consistió en la identificación en tiempo real de la respuesta en frecuencia de la pared arterial de una oveja a partir señales de presión (koninsberg) y diámetro (sonomicrómetro). Se obtuvieron los valores de ajuste buscados y la frecuencia de corte (fc) correspondiente. Este sistema permitirá utilizar la fc en tiempo real para evaluar el estado elástico de las arterias.

Palabras clave— Filtrado adaptivo, Pared Arterial, Señal de Presion, Señal de Diámetro, ADSP-2181

I. INTRODUCCIÓN

La modelización e identificación de sistemas tiene gran importancia en diversos campos de la ingeniería como las comunicaciones, procesamiento de señales y sistemas biológicos [1].

Un filtro adaptivo puede ser utilizado en la modelización imitando el comportamiento de un sistema físico dinámico, el cual es tratado como una “caja negra”, pudiendo tener varias entradas y varias salidas. El modelado de un sistema de entrada y salida únicas puede verse ilustrado en la figura (fig.1).

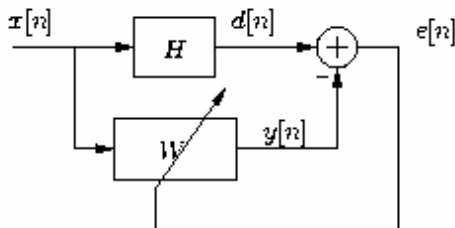


Fig 1: Identificador de sistema adaptivo

El filtro adaptivo ajustará sus coeficientes con el objetivo de que su salida sea igual a la del sistema desconocido. Si el filtro dispone de suficientes coeficientes y estos minimizan el error cuadrático medio entre la salida del filtro y la real, se logrará una buena aproximación y hasta es posible conseguir un ajuste perfecto del sistema.

A partir de la convergencia, la estructura y los valores de los parámetros del sistema adaptivo podrían o no asemejarse a los del sistema a identificar, consiguiéndose la relación entrada-salida buscada. Se puede decir entonces que el sistema adaptivo se convierte en un modelo del sistema a identificar..

La pared arterial puede ser modelada mecánicamente por un sistema de tercer orden [6] en el cual la entrada será la presión y la salida será el diámetro. Es de gran interés poder calcular los parámetros mecánicos del tejido para poder lograr una caracterización de la misma en distintas situaciones fisiológicas; sin embargo en este trabajo se decidió identificar tan solo la respuesta en frecuencia del sistema. Esta representación frecuencial podría ser de utilidad en la detección de posibles patologías en la clínica diaria y aplicarse en tiempo real.

El presente trabajo se dividió en dos etapas; la simulación numérica y la implementación en hardware. La primera de ellas incluyó la realización del algoritmo de identificación de la pared arterial en Matlab ®. Utilizando señales de presión y diámetro reales medidas en animales se buscaron los parámetros óptimos del filtro adaptivo para luego implementar en el hardware. Estos son la velocidad de convergencia (μ), la cantidad de pesos a utilizar en el filtro (L) y el formato numérico que debe utilizarse en el hardware. En la etapa de implementación se realizó el algoritmo en Visual DSP® para poder evaluar su performance frente a la simulación realizada en Matlab ®. Al final de la implementación se programó este algoritmo en un microprocesador especializado para procesamiento de señales ADSP 2181 (Analog Devices) para su utilización y comprobación.

II. MATERIALES Y MÉTODOS

A. Modelización

El combinador lineal, o filtro no recursivo adaptivo, es fundamental para el procesamiento adaptivo de señales. Esto se debe a que es relativamente fácil de entender y analizar. El diagrama de la forma general del combinador lineal adaptivo puede observarse en la figura 1. En donde hayamos:

$$X_k = [x_k \ x_{k-1} \ \dots \ x_{k-L}] \quad (1)$$

siendo X_k el vector de entrada de L muestras, los correspondientes pesos ajustables son el vector columna:

$$W_k = [w_k \ w_{k-1} \ \dots \ w_{k-L}]^T \quad (2)$$

Un sumador y la salida determinada por:

$$y_k = \sum_l^L w_{lk} x_{k-l} \quad (3)$$

$$y_k = X_k^T W_k = W_k^T X_k \quad (4)$$

El nombre combinador lineal proviene de que la salida es una combinación lineal de los componentes de la entrada. Sin embargo, cuando los pesos están en proceso de ajuste, ellos también dependerán de la señal de entrada.

En el proceso de adaptación con realimentación, se ajustan el vector de pesos con el objetivo de conseguir que la salida y_k sea lo mas parecida posible a la señal deseada d_k . Esto se logra minimizando el error que se produce al restar la señal de salida con la señal deseada.

$$\varepsilon_k = d_k - y_k \quad (5)$$

$$\varepsilon_k = d_k - X_k^T W_k \quad (6)$$

B. Modelo Matemático

En nuestro trabajo el proceso de adaptación esta orientado a la minimización del error cuadratico entre la señal deseada a la salida y la salida real del filtro adaptivo. Esto se logra ajustando continuamente los w_k cada vez que se tiene una nueva muestra a la entrada.

Los coeficientes se actualizan utilizando el algoritmo Least Mean Square (LMS), que se basa en el método de descenso por pasos y usa estimaciones “ruidosas” como el gradiente del error cuadrado mínimo[1]. De la ecuación (6) y utilizando el algoritmo LMS, se calcula el gradiente:

$$\begin{aligned} \hat{\nabla}_k &= \frac{\partial \varepsilon^2}{\partial W_k} = 2\varepsilon \frac{\partial \varepsilon}{\partial W_k} \\ &= 2\varepsilon_k \left[\frac{\partial \varepsilon_k}{\partial w_{0k}} \dots \frac{\partial \varepsilon_k}{\partial w_{Lk}} \right]^T \\ &= -2\varepsilon_k X_k \end{aligned} \quad (7)$$

En este mismo algoritmo la actualización de los pesos se calcula a partir de:

$$W_{k+1} = W_k + \mu \hat{\nabla}_k \quad (8)$$

En donde μ es un parámetro de convergencia constante que regula la velocidad y estabilidad de la adaptación.

Reemplazando en las ecuaciones (7) y (8) obtenemos:

$$W_k = W_k + 2\mu \varepsilon_k X_k \quad (9)$$

Como se puede observar el algoritmo LMS es muy simple ya que no se utilizan operaciones complejas. Esta característica permite que su implementación pueda hacerse en tiempo real, pudiendo así obtenerse los coeficientes del filtro que modela la pared arterial, y por lo tanto la respuesta en frecuencia en el mismo momento que se realiza el estudio.

C. Simulación

Las señales utilizadas provienen del estudio realizado por el grupo de Barra, et al. sobre una oveja anestesiada con Acedan y tiopental sódico, y se mantenía en ese estado con Halotano 2%.

Para la señal de presión in vivo se utilizo un transductor de estado solido (Konigsberg), el cual es amplificado y luego digitalizada 10 bits con una frecuencia de muestreo de 4kHz. La medición del diámetro se efectuó con cristales piezoeléctricos, utilizando el sonomicrómetro digitalizada con 10 bits y 4kHz.

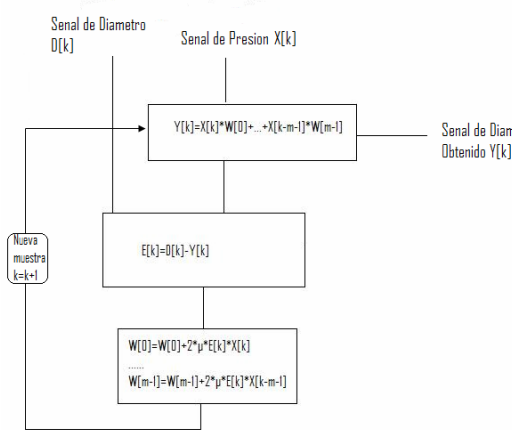


Fig 2: Diagrama en Bloques del sistema

El sistema de numeración utilizado por MATLAB ® en las primeras simulaciones es de punto flotante teniendo así una precisión de 10^{-16} ; recordemos que el microprocesador ADSP 2181 (Analog Devices) utilizado en este trabajo posee una arquitectura de 16 bits de punto fijo, y como se verá más adelante se deberá seleccionar un sistema numérico con el que se obtengan resultados aceptables.

Se realizaron simulaciones en las que se probaron distintos valores de parámetros de convergencia (μ) y cantidad de pesos a utilizar por el filtro. En cada caso se registró el ajuste obtenido y se calculó el error cuadrático medio frente a la señal de diámetro original. En la tabla 1 se pueden observar la performance de cada uno de los filtros simulados para las mismas señales de entrada.

Mu	L	Minimo	Maximo	Error Medio	Varianza
0.5	2	-0.021121993523...	0.1779381336...	-0.00006601182...	0.0000144...
0.001	2	-0.025291159869...	0.1967493095...	0.07385444449953	0.0022745...
0.5	5	-0.017748528776...	0.0740438594...	-0.00017138045...	0.0000062...
0.1	5	-0.023057170238...	0.0736301576...	0.000209540807...	0.0000311...
0.5	10	-20327466.686533...	20334835.204...	-0.00019719908...	8.8098276...
0.05	10	-0.023211987728...	0.0519602590...	0.00026638231295	0.0000356...
0.5	20	-Inf	Inf	NaN	NaN
0.1	20	-0.020066183463...	0.0317189202...	-0.00002599488...	0.0000136...
0.005	50	-0.024792028584...	0.0980311545...	0.001319018002...	0.0001420...
0.001	50	-0.02519404152709...	0.1424180270...	0.004066811910...	0.0004348...
0.005	100	-0.024787075124...	0.0956168444...	0.00137415910991	0.0001375...
0.001	100	-0.025193031604...	0.1421328342...	0.00353001674242	0.0004123...

Tabla 1: Valores simulados

La selección de los valores de μ y cantidad de coeficientes (L) se realizó teniendo en cuenta las siguientes premisas:

- 1) El tiempo que tarda el filtro en ajustar sus coeficientes para lograr la minimización del error cuadrático medio, es llamado "tiempo de establecimiento". Se ha decidido que este tiempo no sea mayor a 20 segundos.
- 2) El error cometido a partir del tiempo de establecimiento no supere el 5% del valor pico de la señal de diámetro obtenida.
- 3) Si frente a dos valores distintos de L la convergencia obtenida es similar, teniendo en cuenta las premisas 1 y 2, se opta por utilizar el L de menor tamaño. Disminuyendo así los tiempos computacionales.
- 4) Si frente a dos valores distintos de μ la convergencia es similar, teniendo en cuenta las premisas 1 y 2, utilizamos el mayor de ambos. Evitamos de esa forma posibles underflows en el código.

Partiendo de estas premisas se procedió a la obtención del μ y de la cantidad de pesos L en señales distintas, evaluando en cada una de ellas la evolución del error en función al tiempo.

Se analizaron seis posibles valores de μ (0,5 – 0,1 – 0,05 – 0,01 – 0,005 – 0,001) manteniendo una cantidad de pesos fijos. En la figura 3, con una L=5 se puede apreciar que el valor μ es directamente proporcional a la velocidad de convergencia.

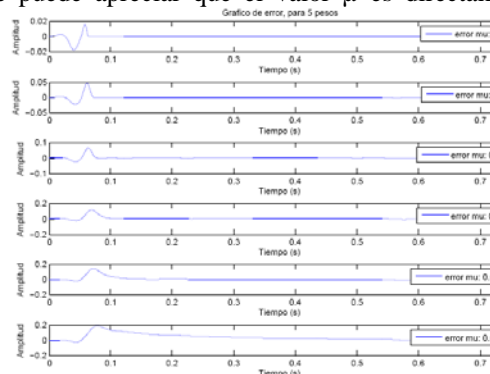


Fig 3: Gráficos de error con distintos μ y L=5, observe que a medida que μ disminuye el tiempo de convergencia aumenta

Utilizando distinta cantidad de pesos (Figura 4) se obtienen convergencias más veloces y menor valor de error. También se destaca que al disminuir mucho el valor de convergencia los tiempos de establecimiento son mayores a los fijados en las premisas.

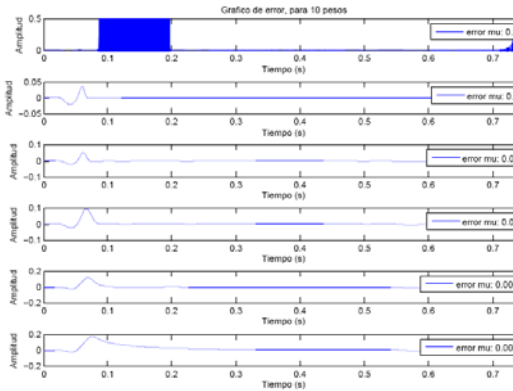


Fig 4: Gráficos de error con distintos μ y $L=10$, observe que a medida que μ disminuye el tiempo de convergencia aumenta

Luego de analizar los resultados obtenidos se opta por seleccionar el parámetros de convergencia igual a $\mu=0.05$. Utilizando este μ se procedió de la misma forma para la obtención del L óptimo. En este caso los valores de L fueron 2, 5, 10, 20, 50 y 100.

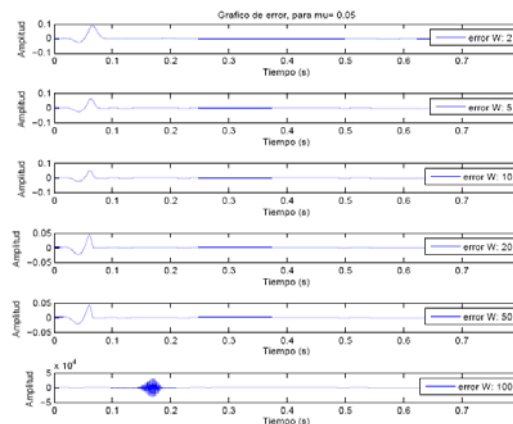


Fig 5: Gráficos de error con distintos L y $\mu=0.05$

Se obtuvieron valores aceptables con poca cantidad de pesos. El hecho de incrementar este valor no represento una disminución susceptible en el error, es por esta razón, y recordando la premisa 3, que seleccionamos $L=10$.

Una vez obtenidos los parámetros constructivos del filtro se procedió a su simulación numérica pensando en el ADSP-2181 recordemos que este posee una arquitectura de 16 bits. Para poder determinar el formato numérico a utilizar se deben tener en cuenta los “overflows” y “underflows” que se generan; estos se deben producir en casos aislados o solo cuando sean permitidos.

Se utilizó el formato numérico Q15, esto significa que se tiene 1 bit de signo y 15 bits de parte fraccional (1.15). La precisión estará representada por el menor valor que se puede obtener, en este caso el valor será de 2^{-15} . Observando nuevamente la tabla 1, comprobamos que los valores no se encuentran por debajo de esta precisión. En las simulaciones realizadas tan solo en el 10 % de las muestras procesadas se encuentran valores por debajo de este valor. En la figura 5b podemos observar la diferencia obtenida entre la señal simulada en punto flotante y la obtenida con formato 1.15.

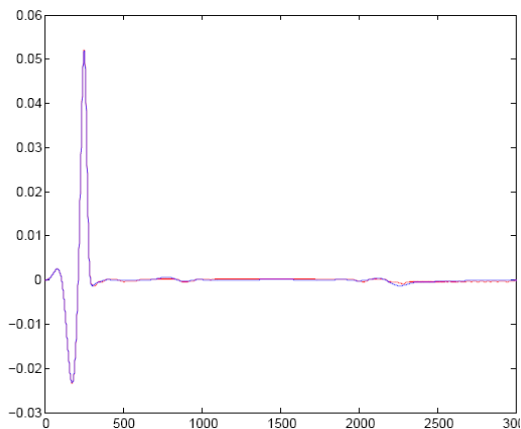


Fig 6: a) Errores obtenidos para Q15 y punto flotante. Se superponen las líneas por cuestiones de escala

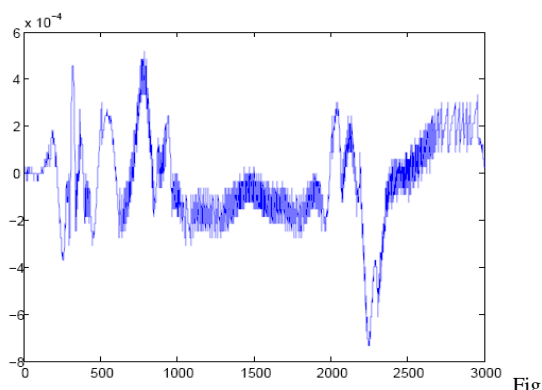


Fig 6: b) Diferencia entre los dos errores, observe el orden de magnitud de 10^{-4} .

Debido a que los algoritmos implementados en el MATLAB son distintos se generó una señal para ser importada por el Visual DSP y se ejecutó el algoritmo offline. Luego se exportó la señal obtenida y los coeficientes del filtro adaptivo desde el espacio de memoria en Visual DSP para poder comparar los resultados con los simulados en MATLAB.

D. Implementación en hardware

El código generado fue diseñado basándose en el hardware disponible y teniendo en cuenta sus limitaciones. El código incluye la programación de 3 bits del puerto entrada/salida (E/S) con el fin de comprobar la frecuencia de muestreo del convertor analógico digital para los dos canales utilizados, otro para el tiempo utilizado por el algoritmo y un último que cambia su estado en el caso de generarse un error al procesar una muestra.

La comprobación del código, se realizó programando el KIT e ingresando las señales de presión y diámetro por la entrada analógica de audio. Se obtuvo así la señal de diámetro generada luego del ajuste del filtro adaptivo. Se analizaron los bits del puerto E/S para corroborar los resultados.

III. RESULTADOS

Se pueden observar en la figura 7 el ajuste realizado por el filtro adaptivo utilizando el μ y L calculado en la simulación, obteniendo así la señal de diámetro.

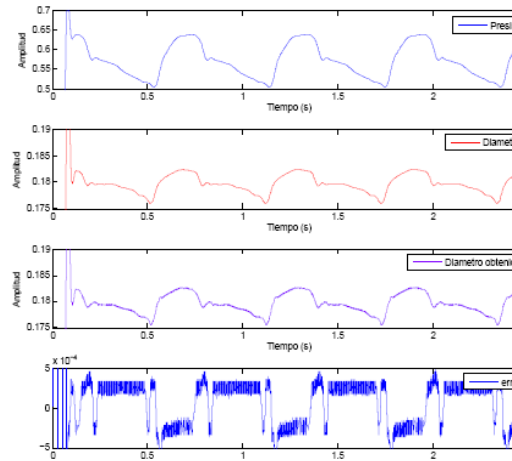


Fig 7: Adaptación del sistema y el error obtenido

Los resultados obtenidos en el hardware difieren de los obtenidos en la simulación computacional. Esto se debe a los filtros pasa bandas (20 Hz a 20 kHz) que posee el kit utilizado (ADSP-2181 EZ Kit-Lite evaluation system) a la entrada y a la salida de los conversores analógico digitales; recordemos que la señal de presión y diámetro posee componentes por debajo de los 20 Hz de gran energía [5].

IV. DISCUSIONES Y CONCLUSIÓN

Se contrastaron los valores obtenidos del filtro adaptivo implementado en el DSP offline con los simulados en el MATLAB, obteniendo resultados muy aceptables en cuanto a la identificación del sistema. Las diferencias obtenidas entre ambas se deben a los formatos numéricos utilizados, en la implementación en hardware el error máximo obtenido tanto en la señales de presión y diámetro como en los pesos del filtro es de 10^{-3} . Los valores de orden del filtro L y el coeficiente de convergencia se ajustaron siguiendo los criterios antes mencionados y resultaron ser $\mu=0.05$ y $L=10$. La respuesta en frecuencia de la pared arterial resulto un filtro pasa bajos de f_c de 70 Hz (Fig. 8).

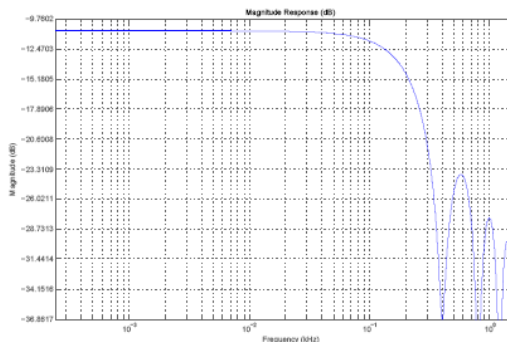


Fig 8: Respuesta en frecuencia del filtro obtenido

El efecto del filtro pasa banda que presenta la salida analógica del kit (ADSP-2181 EZ Kit-Lite evaluation system) provoca que la señal aplicada a la entrada no sea la misma obtenida a la salida. Este factor es de gran importancia para la validación de los resultados, ya que las señales de presión y diámetro que ingresan al filtro adaptivo difieren de los simulados en la PC, es por esta razón que los coeficientes simulados en tiempo real en el KIT no son los mismos que los obtenidos en la simulación en MATLAB. Para solucionar este inconveniente se debe optar por la utilización del DSP fuera del kit que hemos utilizado. En futuros trabajos los coeficientes del filtro serán enviados mediante el puerto serie, para la visualización de la respuesta en frecuencia de la pared arterial.

La utilización de esta herramienta permitirá medir en tiempo real la f_c de la pared arterial.

V. REFERENCIAS

- [1] "Adaptive signal processing", B. Widrow. N.J.: Prentice-Hall,1985, p.56.
- [2] "Adaptive signal processing", B. Widrow. N.J.: Prentice-Hall,1985, p.52.
- [3] MATLAB ® - MathWorks.
- [4] Visual DSP®. Analog Devices
- [5] Adaptive signal processing", B. Widrow. N.J.: Prentice-Hall,1985, p.335-337
- [6] "Identification of arterial wall dynamics in conscious dogs", LG Gamero, Experimental physiology 2001;86:519-528
- [7] "Ingeniería cardiovascular: modelos y aplicaciones", Armentano R.L.