

# Ecualización de señales en un enlace digital de comunicaciones

Rocío Parra  
Lucero Guadalupe Fernandez  
Instituto Tecnológico de Buenos Aires

**Resumen**—Se ecualizó una señal en un enlace digital basándose en el esquema de filtrado de inversión de sistemas. Se implementaron NLMS y RLS, justificando la elección de los parámetros, consiguiendo minimizar el bit error rate, la tasa de errores. Se implementó NLMS, justificando la elección de los parámetros, consiguiendo minimizar el bit error rate, la tasa de errores. Luego se buscó de manera exitosa implementar RLS, consiguiendo aún mejores resultados para el problema.

## I. INTRODUCCIÓN

Se buscó ecualizar una señal en un enlace digital de comunicaciones. Los datos consistían en una secuencia de datos pseudoaleatoria codificada por Manchester, muestreada a una frecuencia de sampleo de 4 kHz a razón de 250 bps. Ante estas características, cada bit consiste de 16 muestras según la codificación mencionada. El canal, en este caso, conocido, modificaba la señal dependiendo del posicionamiento aleatorio de dos pares de polos conjugados. En lo que respecta a estacionaridad, el canal carece de la misma debido a sus características cambiantes en el tiempo. La principal dificultad consistía entonces en la variabilidad del canal. En las siguientes figuras se puede notar el efecto de lo mencionado, siendo la señal en azul los bits enviados y en naranja lo recibido.

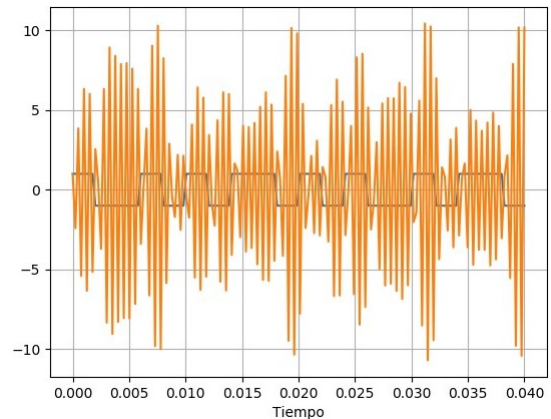
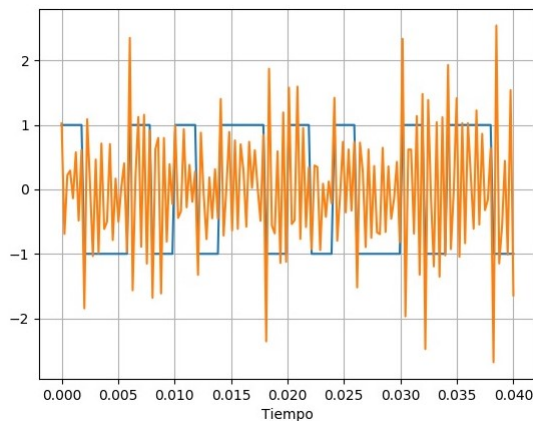
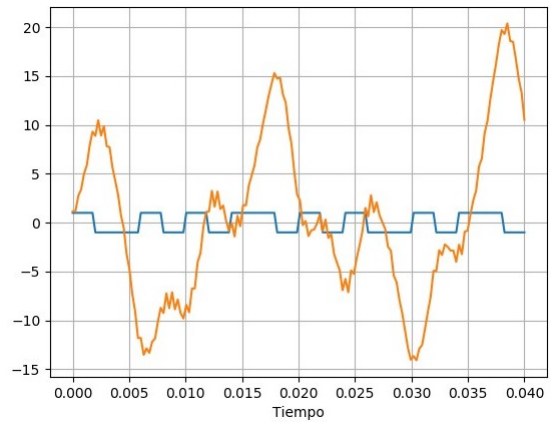
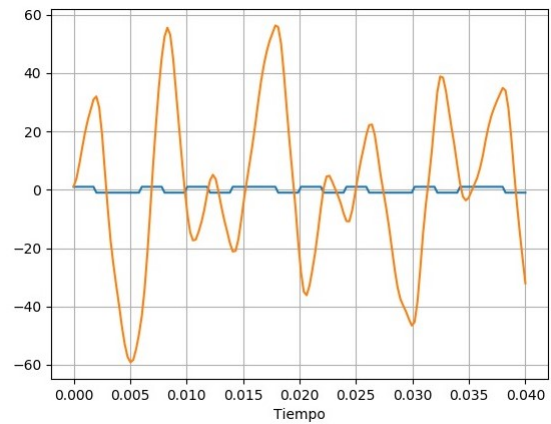


Figura 1. Señales obtenidas en el receptor, al pasar por el canal.

El proyecto consiste en aplicar un algoritmo de filtrado adaptativo para recuperar la señal transmitida asumiendo que no se conoce la entrada, lo que simularía un enlace digital de comunicaciones. Un esquema de lo mencionado se puede ver en la siguiente figura.

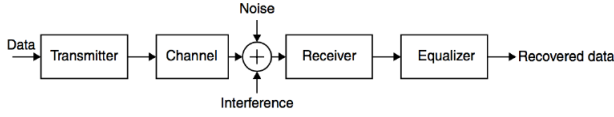


Figura 2. Esquema del enlace digital.

Como criterio de validación del algoritmo como así de sus parámetros, se calculó el bit error rate (BER), la cantidad de bits errados por unidad de tiempo.

## II. ECUALIZACIÓN ADAPTATIVA POR INVERSIÓN DE SISTEMAS

El objetivo del proyecto es invertir los efectos del canal -el sistema desconocido- mediante el esquema de filtrado adaptativo conocido como inversión de sistemas. El diagrama se muestra a continuación:

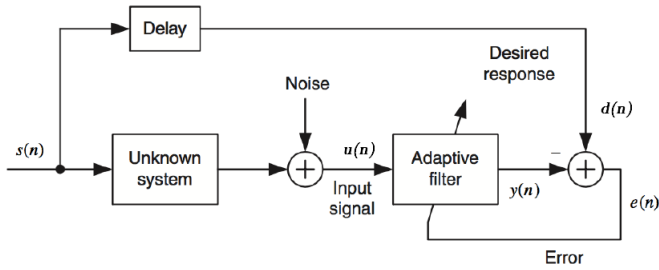


Figura 3. Inversión de sistemas.

En principio se asume que el ruido del canal es incorrelacionado con  $s(n)$ . El filtrado funciona de la siguiente manera, se tiene la señal de entrada  $s(n)$ , que se transmite por el sistema desconocido -el canal ya mencionado-, cuya salida es  $u(n)$ , la entrada al filtro adaptativo, de donde se obtiene  $y(n)$ . Realimentando la señal de error  $e(n) = d(n) - y(n)$  al filtro adaptativo se maximiza la correlación entre la salida del filtro y la señal deseada  $d(n)$ . Opcionalmente, a su vez, se puede colocar un delay, retrasando la señal al obtener  $d(n)$  para compensar el delay propio del sistema. Con ésto se consigue una salida con una respuesta en frecuencia inversa al sistema desconocido, lo que anula su efecto. Sin embargo, en un enlace digital el receptor no conoce la respuesta deseada, por lo que este esquema no es de utilidad. La solución consiste en utilizar una secuencia de entrenamiento, una respuesta deseada  $d(n)$  preacordada entre emisor y receptor. El diagrama de bloques completo se observa en la figura siguiente:

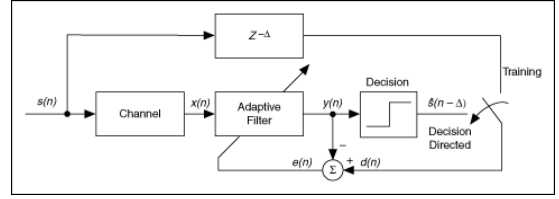


Figura 4. Diagrama de bloques con decision-directed feedback y secuencia de entrenamiento.

Luego del período de entrenamiento inicial los coeficientes del ecualizador pueden ser continuamente ajustados con un decision-directed feedback. De esta manera, la señal de error  $e(n) = d(n) - y(n)$  se deriva del último (no necesariamente correcto) bit estimado de la secuencia transmitida  $u(n)$ .

En este caso, como la señal se encontraba codificada en formato Manchester, la decisión del bit recibido no consistía en un simple comparador, sino que se utilizó la regla de decisión bayesiana [1]. Partiendo de la regla:

$$y^T \cdot (s_1 - s_0) \underset{H_0}{\overset{H_1}{\gtrless}} \sigma^2 \cdot \ln \left( \frac{P(H_0)}{P(H_1)} \right) + \frac{1}{2} (s_1^T s_1 - s_0^T s_0) \quad (1)$$

donde  $y$  es un vector con 16 mediciones,  $s_1$  contiene las 16 muestras que forman un bit de 1 (es decir, 8 veces el valor -1 seguido de 8 veces el valor 1), y  $s_0$ , las que forman un 0. Como  $s_0 = -s_1$ , y ambos símbolos son equiprobables (lo cual asumimos porque no conocemos a priori las características del mensaje que se mandará), la expresión se ve simplificada en:

$$y^T \cdot s_1 \underset{H_0}{\overset{H_1}{\gtrless}} 0 \quad (2)$$

Considerando las características de la codificación Manchester, la regla de decisión resulta ser:

$$\sum_{i=0}^7 y_i \underset{H_0}{\overset{H_1}{\gtrless}} \sum_{i=8}^{15} y_i \quad (3)$$

## III. FILTRO ADAPTATIVO

Al momento de elegir el algoritmo que se implementó para el filtro, se analizaron varias alternativas, entre ellas LMS, NLMS, VS-LMS, Sign LMS y RLS.

En primer lugar, los algoritmos Sign LMS, entre ellos, sign-error, sign-data y sign-sign fueron descartados ya que el baud rate de la señal era de 250bps, cabe recordar que esta variante de LMS es de utilidad para ecualizar canales de comunicación digital de alta velocidad.

En segundo lugar y luego de analizar los resultados obtenidos y las conclusiones propuestas por Bismor [2], se decidió no implementar VS-LMS. En las palabras de los autores: "no hay algoritmo VS-LMS que sea tan versátil, fácil de implementar y adecuado para aplicaciones en tiempo real como el NLMS".

Esta observación, si bien descarta la implementación de VS-LMS, plantea un último debate respecto a si corresponde

utilizar LMS o NLMS. Se consideró entonces, analizar el parámetro de paso que se utilizaría para LMS. Se recuerda que el paso para que converja LMS está dado por

$$0 < \mu < \frac{2}{\lambda_{\text{máx}}} \quad (4)$$

en donde  $\lambda_{\text{máx}}$  es el autovalor más grande de  $R$ , la matriz de autocorrelación de  $u(n)$ . En la medida en que el  $\mu$  elegido se acerque al valor máximo, la velocidad de convergencia aumenta, pero así también el desajuste  $\mathcal{M}$ . Por el contrario, al disminuir el paso, la convergencia se ralentiza, y disminuye  $\mathcal{M}$ . Se trata de una relación de compromiso entre la velocidad de convergencia y el desajuste.

Se realizaron 5000 simulaciones y se calculó el  $\mu_{\text{máx}}$  para cada una de ellas. Se muestra a continuación un histograma con la frecuencia de aparición del paso máximo, en función de este y los  $\mu_{\text{máx}}$  en función del número de simulación.

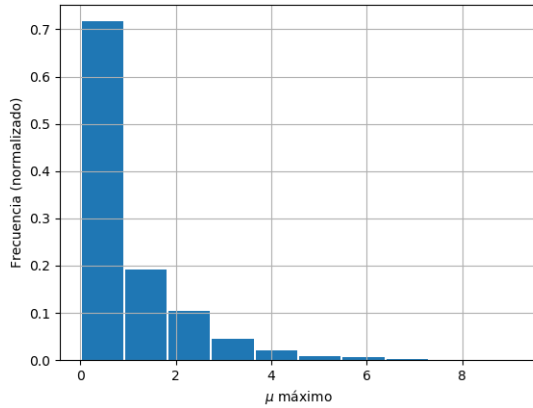


Figura 5. Frecuencias de  $\mu_{\text{máx}}$  en función del mismo.

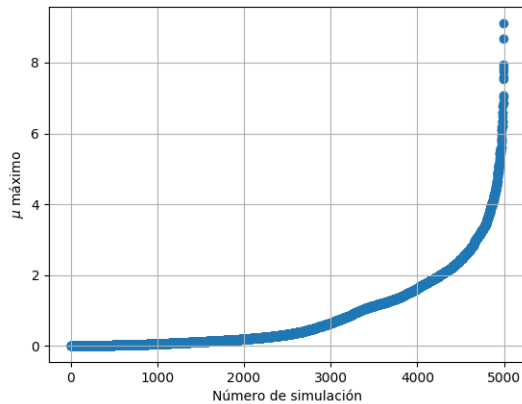


Figura 6.  $\mu_{\text{máx}}$  en función del número de simulación.

El valor del parámetro de paso varía entre valores cercanos a 0 y mayores a 8, ya que la energía de la señal varía en cada realización según el posicionamiento de los polos del canal.

Lo último presenta un problema al elegir el  $\mu$  conveniente para la situación real, puesto que si se elige un paso pequeño, tardará de manera considerable cuando el  $\mu_{\text{máx}}$  sea alto; y si se decide por un  $\mu$  mayor, divergirá en los otros casos.

Debido a lo expuesto anteriormente, se decidió no utilizar LMS, en pos de que no diverja el algoritmo. Esto se logró utilizando NLMS, pues lo que se introduce es un parámetro de paso variable, que depende de la energía de la señal de entrada, por lo que la misma ya no es un problema.

Por otro lado, se implementó a su el filtro con el algoritmo de RLS.

## NLMS

### Implementación

**Orden del filtro:** Como se mencionó en secciones anteriores, el canal es, si bien aleatorio, conocido, por lo que se pudo observar analizándolo que contaba con dos pares de polos conjugados, 4 polos. Resulta entonces trivial la observación que con un filtro FIR de orden 4, los efectos de las singularidades se verían neutralizados. Se implementó entonces NLMS, y se realizó una simulación de Montecarlo del bit error rate (BER) para órdenes de filtro entre 4, 5 y 6, en función del parámetro  $\mu$ .

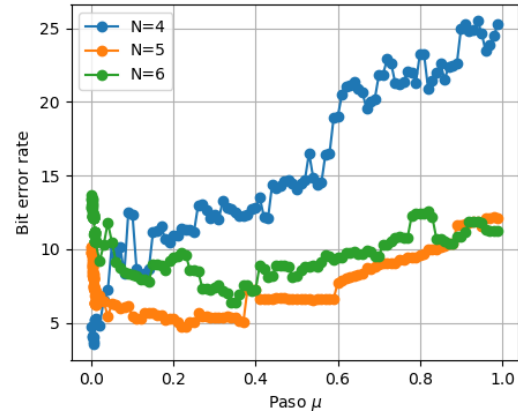


Figura 7. Bit error rate en función de  $\mu$  para orden 4, 5 y 6 del filtro.

Se observa que para valores de  $\mu$  pequeños, el algoritmo mantiene un BER bajo para orden 4, desestabilizándose para pasos más grandes; no así ocurre con orden 5, que si bien tiene un mayor BER para valores bajos, tiene resultados más consistentes para valores mayores de  $\mu$ . A su vez, este orden obtiene mejores resultados que un filtro de sexto orden. Con todo, el orden del filtro elegido fue 5.

**Delay:** En lo que respecta al delay, bloque del esquema de inversión de sistemas, figura 3, el retardo funciona como una compensación del delay real del filtro y del canal. Además, es sólo utilizado para la secuencia de entrenamiento, para obtener la señal deseada. En primer lugar se acotó el valor del delay entre 0-4 por el orden ideal del filtro, luego realizó un

Montecarlo para estos valores de delay, donde nuevamente se graficó el BER en función del parámetro de paso. Obteniendo una mejor tasa de error para el caso de delay unitario por lo que fue elegido.

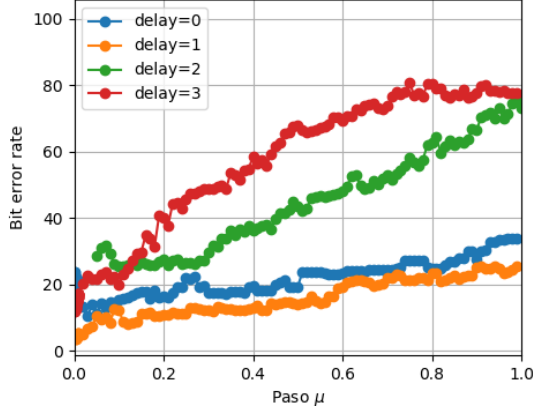


Figura 8. Bit error rate en función de  $\mu$  para delays entre 0-4.

*Paso  $\mu$ :* De manera análoga, con base en ambos gráficos, se eligió un paso  $\mu = 0.005$ ; el valor para el cual tanto para un delay unitario, como para un filtro de quinto orden el BER obtenido era menor al 5 %.

#### RLS

En contraste con LMS se tiene el algoritmo RLS. Le gana en velocidad de convergencia, aunque este beneficio tiene como costo alta complejidad computacional.

En este algoritmo se introduce un factor de peso exponencial  $\beta$ , también conocido como factor de olvido, definido por:

$$\beta(n, i) = \lambda^{n-i} \quad i = 1, 2, \dots, n \quad (5)$$

donde la constante  $\lambda$  vale entre cero y uno. Cuanto más cercano a la unidad valga  $\lambda$ , más valores del pasado pesan. El caso máximo se cumple con  $\lambda = 1$ , donde el algoritmo tendría memoria infinita. Esto resulta de utilidad si el proceso es estacionario.

En segundo lugar, RLS cuenta con otro parámetro conocido como parámetro de regularización,  $\delta$ . A niveles prácticos, la regularización suaviza la solución obtenida ya que castiga los valores altos de  $w(n)$ . La función de costo para este algoritmo es función de ambos parámetros mencionados, entonces de manera análoga al caso del NLMS, se calculó el BER, variando el parámetro de regularización con el fin de encontrar el factor de olvido óptimo para el problema. En tal caso, se realizaron simulaciones de 1000 realizaciones del proceso de 100 bits cada una posteriores a 15 bits de entrenamiento, para órdenes 4, 5 y 6 del filtro para distintos valores del factor de regularización  $\delta$  y factor de olvido  $\lambda$ .

Considerando que la constante de tiempo RLS está dada por:

$$\tau = -T_s \cdot \ln(\lambda) \quad (6)$$

Poniéndolo en términos del número de muestras que representa  $k$  tal que  $\tau = k \cdot T_s$ , obtenemos que:

$$\lambda = e^{-\tau/T_s} = e^{-1/k} \quad (7)$$

De esta manera, se utilizaron valores con espaciado lineal de  $k$  entre un cuarto de bit (4 muestras) y 15 bits (240 muestras). Los resultados obtenidos se ilustran en las figuras 9 a 11.

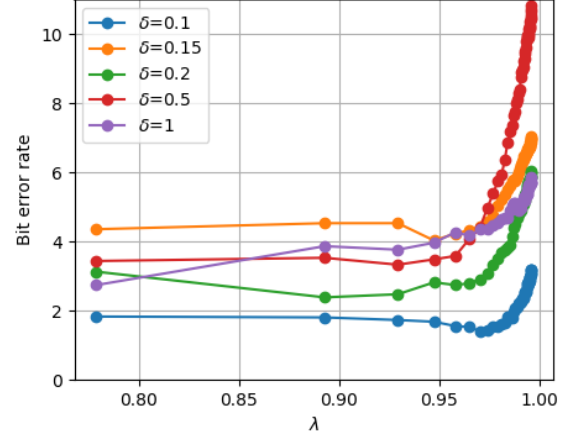


Figura 9. Bit error rate en función de  $\mu$  para distintos valores de  $\delta$ , con  $N = 4$ .

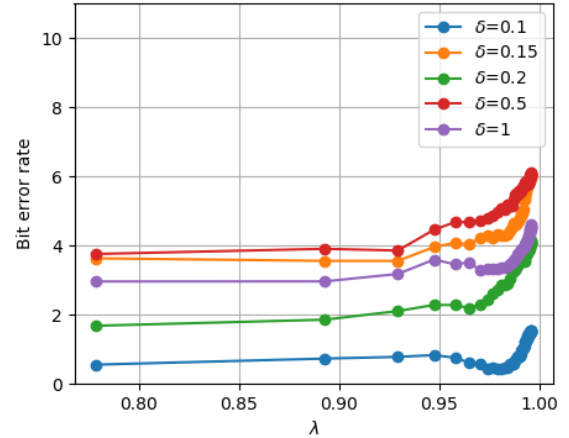


Figura 10. Bit error rate en función de  $\mu$  para distintos valores de  $\delta$ , con  $N = 5$ .

Se observa que el mejor resultado se obtiene nuevamente para  $N = 5$ , es decir sobreajustando el filtro. En este caso, el menor bit error rate se obtiene con  $\delta = 0.1$  para  $\mu \sim 0.9738$ , es decir que  $k$  es aproximadamente 38 muestras, o 2.35 bits.

Con estos parámetros, el bit error rate obtenido es de 0.425, es decir aproximadamente un 0.17 %. Esto es una mejora considerable respecto de NLMS, con el cual el error obtenido era superior a 2 bits por segundo, es decir prácticamente el doble.

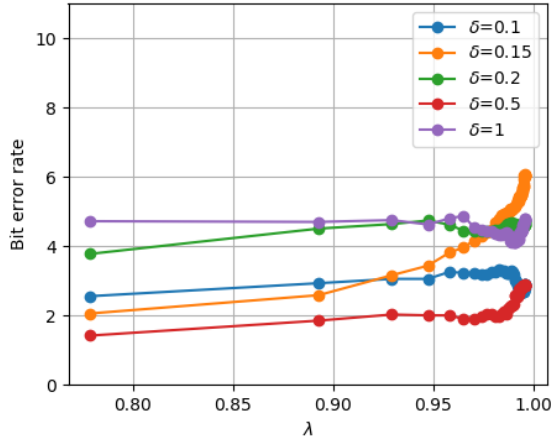


Figura 11. Bit error rate en función de  $\mu$  para distintos valores de  $\delta$ , con  $N = 6$ .

#### IV. CONCLUSIONES

Considerando el análisis previamente realizado, se determinó que la ecualización óptima del canal se logra con un esquema de inversión de sistema, actualizando los coeficientes del filtro adaptativo mediante el método RLS. Los parámetros óptimos de este procedimiento se determinaron como:

$$\begin{aligned} N &= 5 \\ \lambda &= 0.9738 \\ \delta &= 0.1 \end{aligned} \quad (8)$$

El método recomendado es, pues:

1. **Entrenamiento del canal:** la señal deseada es conocida, y es la entrada con 1 tap de delay (para compensar el delay introducido por el sistema)
2. **Transmisión:**
  - a) utilizando el último valor de  $w$ , se obtienen 16 valores de  $y$
  - b) se realiza el test expresado en la ecuación 3
  - c) según el resultado del test, se obtiene el nuevo valor de  $w$ , procesando nuevamente las últimas 16 muestras, pero esta vez utilizando  $d = s_0$  o  $d = s_1$  según corresponda

El bit error rate obtenido en estas condiciones es de 0.17 %. De no ser esto suficiente para una comunicación exitosa, se debería utilizar una codificación con detección y corrección de errores (por ejemplo, Hanning).

#### REFERENCIAS

- [1] K. S. Shanmugan, A. M. Breipohl. *Random Signals: Detection, Estimation and Data Analysis*. Wiley, May 1988.
- [2] D. Bismor, K. Czyz and Z. Ogonowski. "Review and Comparison of Variable Step-Size LMS Algorithms". *International Journal of Acoustics and Vibration* vol. 21, no. 1, pp. 24-39, 2016.
- [3] S. Qureshi. "Adaptive Equalization". *IEEE Communications Magazine*, vol. 20, no. 2, pp. 9-16, March 1982.
- [4] S. Qureshi. "Adaptive Equalization". *Proceedings of the IEEE*, vol. 73, no. 9, pp. 1349-1387, 1985.

- [5] National Instruments. *Adaptive Channel Equalization (Digital Filter Design Toolkit)*, 2014. [Online]. Disponible: [http://zone.ni.com/reference/en-XX/help/371988G-01/lvaftconcepts/aft\\_ace/](http://zone.ni.com/reference/en-XX/help/371988G-01/lvaftconcepts/aft_ace/). [Consultado: 11-Nov-2019].