

**ANALIZADOR DIGITAL AUTOMÁTICO DE DISTORSIÓN**



**DIANA MARCELA RUEDA MANTILLA**

**PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA ELECTRÓNICA  
BOGOTÁ, D.C.**

**2010**

**ANALIZADOR DIGITAL AUTOMÁTICO DE DISTORSIÓN**

**DIANA MARCELA RUEDA MANTILLA**

Trabajo de grado presentado como  
requisito parcial para optar al título  
de Ingeniera Electrónica.

Director:

**PEDRO RAÚL VIZCAYA GUARÍN Ph.D.**  
Ingeniero Electrónico

**PONTIFICIA UNIVERSIDAD JAVERIANA**  
**FACULTAD DE INGENIERÍA**  
**CARRERA DE INGENIERÍA ELECTRÓNICA**  
**BOGOTÁ, D.C.**

**2010**

**PONTIFICIA UNIVERSIDAD JAVERIANA**

**FACULTAD DE INGENIERÍA**

**CARRERA DE INGENIERÍA ELECTRÓNICA**

RECTOR MAGNÍFICO: R.P. JOAQUIN EMILIO SANCHEZ GARCÍA S.J

DECANO ACADÉMICO: Ing. FRANCISCO JAVIER REBOLLEDO MUÑOZ

DECANO DEL MEDIO UNIVERSITARIO: R.P SERGIO BERNAL RESTREPO S.J

DIRECTOR DE CARRERA: Ing. JUAN MANUEL CRUZ BOHÓRQUEZ M, Ed.

## **NOTA DE ADVERTENCIA**

“La Universidad no se hace responsable de los conceptos emitidos por algunos de sus alumnos en los proyectos de grado. Solo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vea en ello el anhelo de buscar la verdad y la justicia.”

*Artículo 23 de la Resolución No. 13, del 6 de julio de 1946, por la cual se reglamenta lo concerniente a Tesis y Exámenes de Grado en la Pontificia Universidad Javeriana.*

## **AGRADECIMIENTOS**

Agradezco inicialmente a mis padres y a mi familia por apoyarme durante el desarrollo de este trabajo de grado. Sus palabras alentadores me ayudaron a superar varias adversidades.

Expreso mi gratitud a mi director, el Ingeniero Pedro Raúl Vizcaya, por haber sido una guía incondicional durante todo el proceso del trabajo de grado. Asimismo, aprecio profundamente las ayudas y recomendaciones brindadas por mi asesor, el Ingeniero Camilo Otálora, y por mi compañero el Ingeniero Santiago Valencia.

Adicionalmente, agradezco la asistencia, paciencia y eficacia en el servicio brindado por el Laboratorio de Ingeniería Electrónica.

Finalmente, reconozco la gran labor y amabilidad mostradas por los celadores y el personal de aseo de la facultad.

## **DEDICATORIAS**

*Dedico este trabajo de grado a mi familia que siempre ha velado por mí y que quiero infinitamente, y a todos los profesionales, amigos y familiares que me ayudaron a recobrar mi buena salud.*

*DIANA MARCELA RUEDA MANTILLA*



## TABLA DE CONTENIDO

1	GLOSARIO.....	2
2	INTRODUCCIÓN .....	3
3	MARCO TEÓRICO.....	5
	3.1 <i>Conversores A/D Sigma Delta</i> .....	5
	3.2 <i>Filtrado adaptable</i> .....	7
	3.3 <i>Medidas de ruido y distorsión</i> .....	13
	3.4 <i>Cálculo de la varianza</i> .....	14
4	ESPECIFICACIONES .....	16
5	DESARROLLOS .....	18
	5.1 <i>Simulaciones</i> .....	18
	5.2 <i>Implementación en hardware</i> .....	19
	5.3 <i>Algoritmos de implementación</i> .....	21
6	ANÁLISIS DE RESULTADOS .....	26
	6.1 <i>Análisis piso de ruido</i> .....	26
	6.2 <i>Variación frecuencia señal de entrada (fi)</i> .....	28
	6.3 <i>Variación frecuencia de muestreo (fs)</i> .....	29
	6.4 <i>Variación número de pesos del filtro (N)</i> .....	32
	6.5 <i>Valores máximos</i> .....	34
	6.6 <i>Prueba integrada 1</i> .....	37
	6.7 <i>Prueba integrada 2</i> .....	43
7	CONCLUSIONES .....	50
8	BIBLIOGRAFÍA Y FUENTES DE INFORMACIÓN .....	52

## 1 GLOSARIO

SINAD (*Signal to Noise and Distortion Ratio*): Relación señal a ruido mas distorsión.

THD (*Total Harmonic Distortion*): Distorsión armónica total

THD + N (*Total Harmonic Distortion plus Noise*): Distorsión armónica total más ruido.

SNR (*Signal to Noise Ratio*): Relación señal a ruido.

ADDA (*Automatic Digital Distortion Analyzer*): Analizador digital automático de distorsión.

PCM (*Pulse Code Modulation*): Modulación por impulsos codificados.

A/D, ADC (*Analog to Digital Conversion*): Conversión analógico a digital.

D/A (*Digital to Analog Conversion*): Conversión digital a analógico.

LMS (*Least Mean Squares algorithm*): Algoritmo de los mínimos cuadrados.

FFT (*Fast Fourier Transform*): Transformada rápida de Fourier.

DSP (*Digital Signal Processing*): Procesamiento digital de señales

ENOB (*Effective Number of Bits*): Número efectivo de bits.



## 2 INTRODUCCIÓN

La idea de un Analizador Digital de Distorsión se generó inicialmente gracias a la necesidad de medir el ruido y la distorsión en simulaciones en Simulink<sup>1</sup> para los cursos de Comunicaciones impartidos en la Universidad. El Ing. Pedro Vizcaya utilizó en sus simulaciones un bloque que diseñó basándose en parte en una variación de la estructura adaptable de cancelación de ruido. Al notar que el bloque analizador digital de distorsión se comportaba de una manera satisfactoria en las simulaciones, se pensó en la posibilidad de implementarlo en hardware.

En la universidad existen varios instrumentos para medir la distorsión, uno de esos es el Leader Idm-171 [1](Medidor de distorsión semi-automático) que está basado en el método de “análisis de distorsión armónica por supresión de la frecuencia fundamental” [2] . Entre las desventajas de este instrumento está el requerimiento de la manipulación constante de una persona durante la realización de mediciones; un ejemplo es el uso del vernier para eliminar la componente de la frecuencia fundamental durante las mediciones de la distorsión armónica total. La limitación del ajuste manual puede llevar a mediciones inexactas y asimismo a confusión en los usuarios que no se han familiarizado con el instrumento.

Actualmente existen analizadores digitales de distorsión pero de una gama relativamente costosa, algunos son usados como herramientas de ayuda para manufactura, desarrollo y control de calidad de productos de audio. Un ejemplo de estos analizadores digitales es el *Klippel R&D System* [3] , que tiene algunas características, como una unidad de procesamiento digital con una interfaz para el usuario, operación de dispositivo autónomo y operación controlada por computador, entre otras.

Otro de los instrumentos encontrados fue el *Sencore DigiPro Digital Audio Analyzer*. [4] Este dispositivo tiene características bastante satisfactorias pero al elevado costo de \$4,495.95 dólares. [5]

También está el instrumento diseñado por Philips Research Labs., Eindhoven [6] , Holanda. Se llama *Real-Time Digital Distortion Analyzer*<sup>2</sup>. Este analizador utiliza un detector de frecuencia para controlar un filtro *notch* que remueve la señal seno de prueba o el *twin-tone* entre 10Hz y 20KHz. La distorsión armónica total y el ruido amplificado restante son calculados y aplicados a un monitor ADC<sup>3</sup>. Esta implementación no es realmente digital porque el filtrado es analógico y la sección digital se encuentra al final para entregar la medida en formato digital. Además, este instrumento utiliza el mismo principio que los analizadores de distorsión analógicos a diferencia del analizador digital automático de distorsión desarrollado en este trabajo de frado.

Al implementar el instrumento en hardware se buscó que fuera un dispositivo autónomo, que pueda entregar una medida de la relación señal a ruido más distorsión (SINAD) en tiempo real para análisis de señales de audio. Para lograr desarrollar este instrumento se tuvieron que cumplir tres etapas fundamentales: La conversión analógico-digital de las señales de entrada al instrumento, El filtrado adaptable de las señales digitalizadas, y la estimación del valor de SINAD.

No sólo se cumplieron estas tres etapas, sino que el instrumento entrega el porcentaje de ruido y distorsión de la señal de prueba y también da la opción de observar las señales a la salida del filtrado adaptable en tiempo real.

---

<sup>1</sup> Simulink es una herramienta de programación visual, que funciona sobre el entorno de programación Matlab.

<sup>2</sup> Analizador de distorsión en tiempo real.

<sup>3</sup> ADC por su significado en inglés *Analog to Digital Converter*, que en español sería Conversor Análogo a Digital

Asimismo, el instrumento logra medir la distorsión de señales que están compuestas de una suma de frecuencias, entregando así el valor de SINAD debido al ruido, la distorsión y al efecto de la intermodulación.

El instrumento Analizador digital automático de distorsión será referido en todo el informe como ADDA<sup>4</sup>.

---

<sup>4</sup> Analizador digital automático de distorsión.

### 3 MARCO TEÓRICO

#### 3.1 Conversores A/D Sigma Delta

La conversión Sigma-Delta se generó como una respuesta ante las dificultades y limitaciones que comprende el método tradicional de conversión PCM<sup>5</sup>. Un sistema PCM representa la onda analógica como una señal en amplitud, guardando la información que mide la amplitud muestra por muestra; este proceso de cuantificación introduce errores diferenciales no-lineales en la representación de amplitud. Asimismo, el error total de la conversión es difícil de corregir ya que se utilizan varios bits para representar la onda y cada bit tiene un error diferente a los demás, variando así el error total con cada muestra[9] .

Un conversor Sigma-Delta utiliza 1 sólo bit para codificar toda la señal (en algunos casos, se usan conversores Sigma-Delta con codificación multi-bit hasta de cuatro bits, el rango dinámico aumenta, pero se debe tener en cuenta la velocidad de operación). Para lograr esto, se hace uso de la frecuencia del cambio de la señal. De esta manera, mientras un conversor PCM interpreta la señal como múltiples pasos en amplitud, un conversor Sigma-Delta mantiene los cambios en amplitud constantes y divide la señal en tiempo.

Un esquema general del un modulador Sigma-Delta, con sus respectivos codificador y decodificador, sería el siguiente:

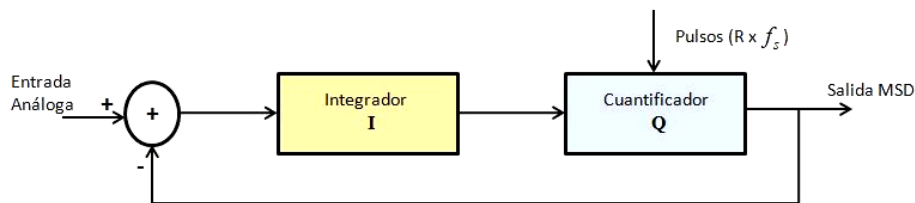


Figura 1. Codificador Modulador Sigma-Delta

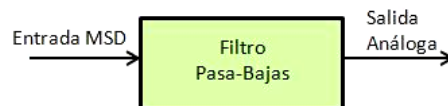


Figura 2. Decodificador Modulador Sigma-Delta

El codificador de la figura 1, es de primer orden. Al cuantificador entra la integral de la diferencia entre la entrada y la salida del cuantificador. El integrador actúa como un filtro pasa-bajas, logrando una retroalimentación en baja frecuencia, ayudando así a que el ruido de cuantificación en bajas frecuencias disminuya. Un sistema Sigma-Delta cuantifica la diferencia (delta) entre la señal actual y la suma (sigma) de la diferencia anterior.

El decodificador sólo requiere un filtro pasa-bajas para remover componentes de alta frecuencia. En general lo que hace es promediar la señal de salida para producir una onda analógica.

<sup>5</sup> Por sus siglas en inglés *Pulse-Code Modulation*

Para la conversión analógico a digital en el instrumento a diseñar se va a utilizar un conversor A/D Sigma-Delta, por eso es importante entender la dinámica de este tipo de conversión, sobretodo sus características en frecuencia.

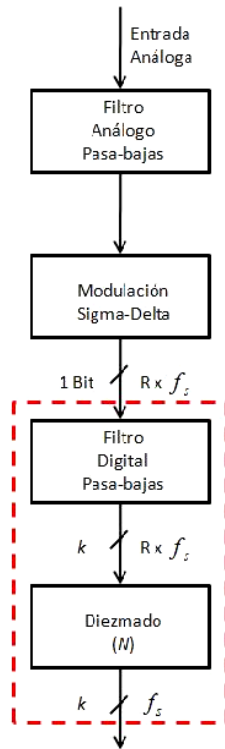


Figura 3. Dinámica en frecuencia de un conversor Sigma-Delta de un Bit.

El filtro analógico pasa-bajas a la entrada del sistema es simple, máximo de segundo orden para asegurar linealidad de fase en la banda de frecuencias de audio. El modulador Sigma-Delta, en este caso de 1 Bit, entrega el bit de información con la frecuencia de muestreo  $f_s$  sobre-muestreada en el factor  $R$ . Los dos últimos bloques hacen parte de la etapa de filtrado de diezmado; en esta etapa se realiza un filtrado pasa bajas del ruido, recortando el espectro para así evitar la aparición de componentes de solapamiento<sup>6</sup> cuando se reduzca la frecuencia de muestreo en el siguiente bloque de diezmado. La función del bloque de diezmado es reducir la tasa binaria y llevarla a valores más manejables o a valores estándar. El factor  $k$  de la gráfica sería el número de bits por muestra.

Cabe mencionar, que los conversores que utilizan sobre-muestreo logran un aumento de resolución (o rango dinámico) no por medio de una disminución del error entre la entrada analógica y la salida digital, sino haciendo que el error ocurra con mayor frecuencia. De este modo, el espectro del error se corre más allá de la banda de audio y aunque el poder del ruido total es alto, el poder de ruido en la banda de audio es bajo.

<sup>6</sup> En inglés este término se define *aliasing*.

### 3.2 Filtrado adaptable

La etapa más importante del Analizador Digital de Distorsión es la etapa de filtrado adaptable. Para lograr este filtrado se hace uso de un filtro llamado *Linear adaptive combiner* que es implementado utilizando el algoritmo LMS<sup>7</sup> que es un algoritmo de búsqueda utilizado para ajustar los coeficientes variables del filtro.

El algoritmo LMS se basa en la señal de error para modificar los coeficientes (o pesos) del filtro adaptable. La estructura básica de un filtro adaptable es la siguiente:

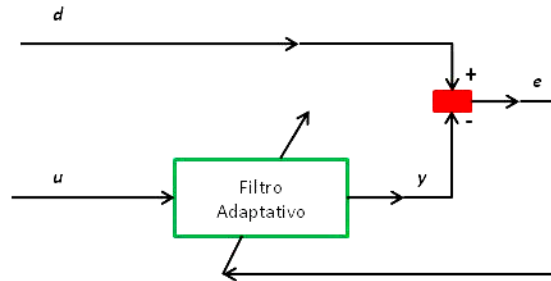


Figura 4. Estructura Básica Filtrado Adaptable.

Esta estructura describe la mayoría de los esquemas adaptables. Las señales, junto con una pequeña descripción, son las siguientes:

$d$ : señal deseada

$u$ : señal de entrada al filtro adaptable

$y$ : señal de salida del filtro adaptable

$e$ : señal de error

Siendo la señal de error:  $e = d - y$

La señal de salida  $y$  es comparada con la señal deseada  $d$  para generar la señal de error  $e$ , que luego es realimentada al filtro adaptable. La señal de error  $e$  es una entrada del algoritmo adaptable, siendo uno de los parámetros que ayuda a la variación de los pesos del filtro adaptable.

#### Estructuras adaptables

Existen varias estructuras adaptables, y cada una varía dependiendo de la aplicación en la cual se va a utilizar el filtrado. En el problema particular de este trabajo, se utiliza una variación de la estructura para *Cancelación de ruido*<sup>8</sup>. Esta variación va a ser denominada: *Cancelación de fundamental*.

<sup>7</sup> Por sus siglas en inglés *Least Mean Squares algorithm*.

<sup>8</sup> En inglés sería *Noise Cancellation*.

## Cancelación de ruido

En la siguiente tabla se van a mostrar las características más importantes de esta estructura.

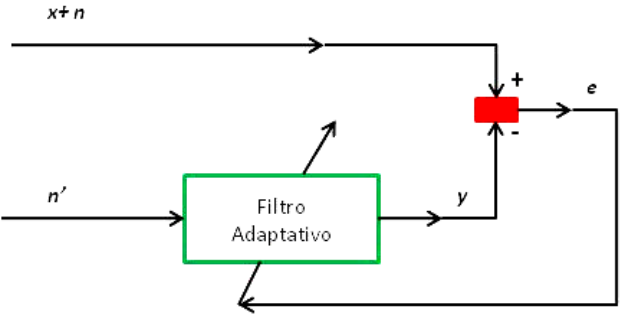
	<i>Estructura cancelación de ruido</i>
<i>Diagrama</i>	
<i>Descripción</i>	<p>La señal de error sería:</p> $e = (x + n) - y$ <p>La señal deseada <math>x</math> tiene ruido aditivo no correlacionado <math>n</math>. La entrada al filtro adaptable <math>n'</math> puede venir de la misma fuente que <math>n</math> pero estaría modificada por el ambiente (<math>n</math> y <math>n'</math> estarían correlacionadas). La salida del filtro adaptable <math>y</math> se va a adaptar a la señal de ruido <math>n</math>, logrando así que la señal de error <math>e</math> tienda a la señal <math>x</math>.</p> <p>En este caso la señal de salida del sistema es <math>e</math> y no <math>y</math>. Esta señal sería la señal deseada sin contenido de ruido.</p>

Tabla 1. Estructura Cancelación de Ruido.

## Cancelación de fundamental

Se va a mostrar una tabla con el mismo formato de la tabla de la estructura de *Cancelación de ruido*, para observar las características más importantes de la estructura de *Cancelación de fundamental* y notar su similitud con la estructura anterior.

<i>Estructura cancelación de fundamental</i>	
<b>Diagrama</b>	
<b>Descripción</b>	<p>La señal de error sería:</p> $e = (x + n) - y$ <p>La señal deseada <math>x</math> tiene ruido y distorsión que va ser representado por la señal <math>n</math>. La entrada al filtro adaptable <math>x'</math> puede venir de la misma fuente que <math>x</math>, pero estaría modificada por el ambiente (<math>x</math> y <math>x'</math> estarían correlacionadas). Para fines de la medida, la señal <math>x'</math> (señal de referencia) va a ser tomada antes que la señal <math>x</math> se distorsione o va generarse con un <i>Estimador de Frecuencia Fundamental</i> que se va aplicar a la señal de entrada al sistema, que en este caso sería la señal distorsionada a ser analizada: <math>x + n</math>.</p> <p>La salida del filtro adaptable <math>y</math> se va a adaptar a la señal sin ruido <math>x</math>, logrando así que la señal de error <math>e</math> tienda a la señal <math>n</math>.</p> <p>De esta manera, la salida del sistema será <math>e</math> y va a entregar el valor de ruido y distorsión en la señal <math>x</math>.</p>

Tabla 2. Estructura Cancelación de Fundamental.

## Algoritmo LMS

El algoritmo LMS fue inventado en 1960 por el profesor de la Universidad de Standord, Bernanrd Widrow y su estudiante de Ph.D. Ted Hoff. Este algoritmo hace parte de la familia de los *algoritmos de gradiente estocástico* (utiliza estimación para hallar el gradiente), distinguiéndose del método de “*steepest descent*” que utiliza un gradiente determinista para el cálculo recursivo del filtro de Wiener para entradas aleatorias. Gran parte de la importancia del algoritmo LMS radica en su simplicidad y facilidad de computación[12] .

Existen dos bloques constitutivos para el filtrado con el algoritmo LMS y pueden ser apreciados en la figura 5:

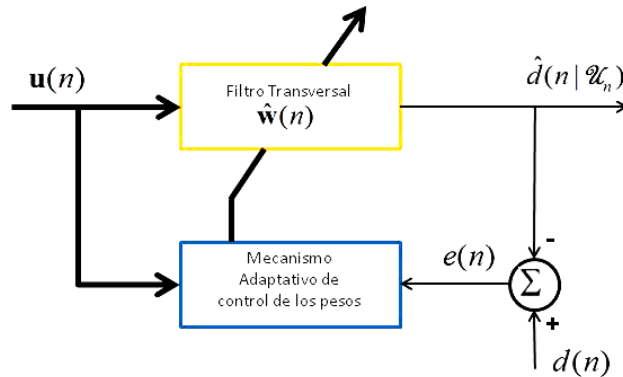


Figura 5. Representación en Bloques Algoritmo LMS.

En el *Filtro Transversal* se cumplen dos funciones principales: Se calcula la salida del filtro transversal, y se compara esta salida con la señal deseada, para así generar una estimación del error. Una gráfica más detallada de este bloque es mostrada a continuación en la Figura 6:

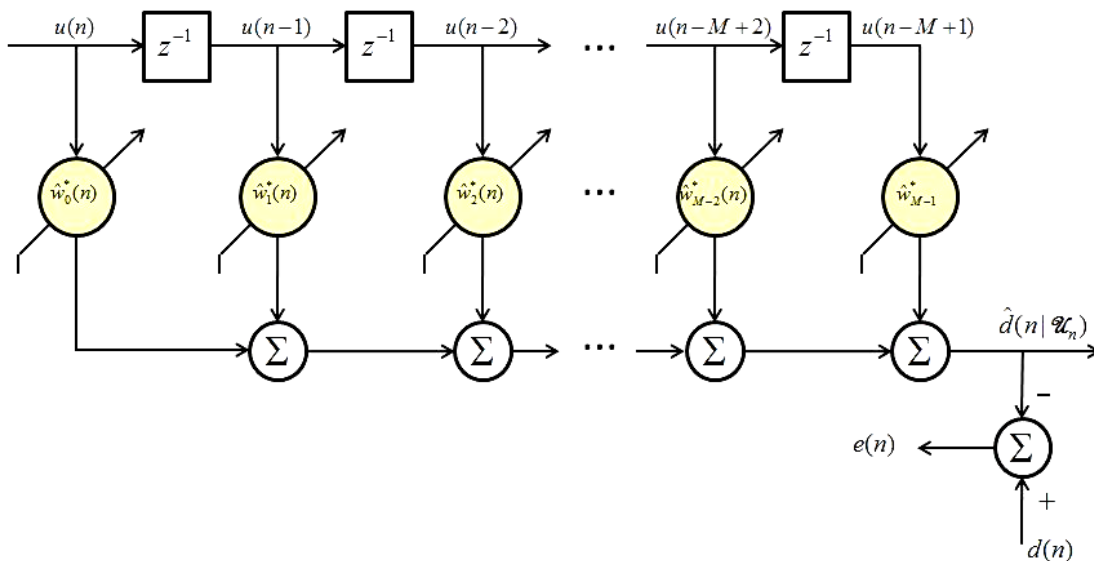


Figura 6. Diagrama de Bloques del Filtro Transversal.

Las entradas para cada etapa del filtro serían  $u(n), u(n-1), \dots, u(n-M+1)$ , formando el vector de entrada  $\mathbf{u}(n)$ , donde  $M$  es el número de etapas del filtro y  $M-1$  es el número de elementos de retraso; estas entradas abarcan un espacio multidimensional representado por  $\mathcal{X}_n$ . Asimismo, los pesos o coeficientes del filtro  $\hat{w}_0(n), \hat{w}_1(n), \dots, \hat{w}_{M-1}(n)$  forman el vector de pesos  $\hat{\mathbf{w}}(n)$ . El valor del vector de pesos,  $\hat{\mathbf{w}}(n)$  calculado usando el filtro LMS, se toma como un valor estimado cuyo



valor esperado se aproxima a la solución óptima de Wiener  $\mathbf{w}_0$  (en un amplio sentido estacionario) a medida que  $n$  se aproxima a infinito.

A la salida del filtro transversal se genera la señal  $\hat{d}(n | \mathcal{X}_n)$  que es usada como un estimado de la respuesta deseada  $d(n)$ . Asimismo, se tiene el error estimado que se toma como la diferencia entre la salida del filtro y la respuesta deseada. La señal de entrada y la señal de error alimentan el siguiente bloque, cerrando así el lazo cerrado de realimentación del filtrado LMS.

Como segundo bloque constitutivo del filtrado LMS está el *Mecanismo Adaptable de control de los pesos* que puede ser apreciado en mayor detalle en la siguiente Figura 7:

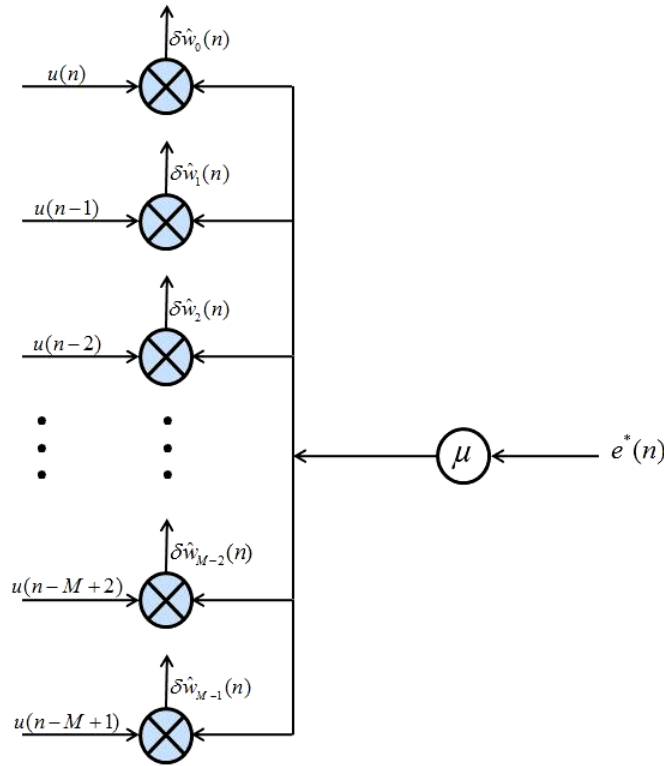


Figura 7. Diagrama de Bloques del Mecanismo Adaptable de Control de los Pesos.

Esta gráfica ilustra cómo se calcula el producto escalar entre el error estimado  $e(n)$  y la entrada  $u(n-k)$  para  $k = 0, 1, 2, \dots, M-2, M-1$ . Cabe notar que existe un factor de escala y este es  $\mu$  que es llamado *parámetro de tamaño de paso*<sup>9</sup>.

Habiendo visto los bloques constitutivos de la etapa de filtrado LMS, ahora se van a enunciar los principios más generales de este algoritmo, y las ecuaciones más importantes que describen su comportamiento.

La ecuación del vector gradiente para hacer que el vector de pesos converja a la solución óptima de Wiener  $\mathbf{w}_0$ , es la siguiente:

<sup>9</sup> En inglés este parámetro es llamado: *step-size parameter*.

$$\nabla J(n) = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}(n) \quad (1)$$

Para hallar este vector gradiente sería necesario tener conocimiento previo de la matriz de correlación de los datos de entrada y la cross-correlación entre las entradas y la respuesta deseada (todos los datos anteriores para todas las etapas del filtro). Para resolver este problema, se toman estimados de  $\mathbf{R}$  y  $\mathbf{p}$  que son definidos de la siguiente manera:<sup>10</sup>

$$\hat{\mathbf{R}}(n) = \mathbf{u}(n)\mathbf{u}^H(n) \quad (2)$$

y

$$\hat{\mathbf{p}}(n) = \mathbf{u}(n)d^*(n) \quad (3)$$

Para definir así el estimado del vector gradiente:

$$\hat{\nabla}J(n) = -2\mathbf{u}(n)d^*(n) + 2\mathbf{u}(n)\mathbf{u}^H(n)\hat{\mathbf{w}}(n) \quad (4)$$

Ahora, substituyendo la ecuación del gradiente estimado, en la ecuación para calcular el peso siguiente (basado en el peso actual) del algoritmo de “**steepest descent**”

$$w(n+1) = w(n) + \frac{1}{2}\mu[-\nabla J(n)] \quad (5)$$

Se llega a la relación recursiva para actualizar el estimado del vector de pesos, que es utilizado en el algoritmo LMS:

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu\mathbf{u}(n)[d^*(n) - \mathbf{u}^H(n)\hat{\mathbf{w}}(n)] \quad (6)$$

Para exponerlo de una manera más clara, acá están las tres ecuaciones principales del algoritmo LMS:

Salida del Filtro:

$$y(n) = \hat{\mathbf{w}}^H(n)\mathbf{u}(n) \quad (7)$$

Error de estimación:

$$e(n) = d(n) - y(n) \quad (8)$$

Adaptación de pesos:

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu\mathbf{u}(n)e^*(n) \quad (9)$$

---

<sup>10</sup> El operador H simboliza la transpuesta conjugada (para números complejos).

### 3.3 Medidas de ruido y distorsión

Existen varias especificaciones para referirse a la cantidad de ruido y/o distorsión en una señal. En la siguiente tabla se van a enumerar las que son más relevantes para este proyecto. Cabe recordar que SINAD es la medida que va a entregar el analizador digital automático de distorsión.

	<b>Distorsión Armónica Total (THD)<sup>11</sup></b>	<b>Distorsión Armónica Total más Ruido (THD + N)<sup>12</sup></b>	<b>Relación Señal a Ruido (SNR)<sup>13</sup></b>	<b>Relación Señal a Ruido más Distorsión (SINAD)<sup>14</sup></b>
<b>Definición</b>	Relación entre el valor rms de la señal fundamental y la raíz cuadrada de la suma de los cuadrados de sus armónicos (generalmente se toman los 5 primeros armónicos como los más significativos).	Relación entre el valor rms de la señal fundamental y la raíz cuadrada de la suma de los cuadrados de sus armónicos más los cuadrados de todas las componentes de ruido (excluyendo dc). Para esta medida se debe especificar el ancho de banda. Para la FFT el ancho de banda es desde dc hasta $f_s/2$ (el ancho de banda de Nyquist), en este caso las medidas de THD + N y SINAD son las mismas.	Relación entre el valor rms de la señal fundamental y la raíz cuadrada de la suma de los cuadrados de todos los otros componentes del espectro, excluyendo los armónicos, y excluyendo el dc.	Relación entre el valor rms de la señal fundamental y la raíz cuadrada de la suma de los cuadrados de todos los otros componentes del espectro, incluyendo los armónicos, y excluyendo el dc.  A veces también se refiere como $S/(N + D)$ .

Tabla 3. Tabla comparativa de las medidas de Ruido y Distorsión.

<sup>11</sup> Las siglas THD se refieren a Distorsión Armónica Total en inglés (*Total Harmonic Distortion*).

<sup>12</sup> Se refiere a *Total Harmonic Distortion plus Noise*.

<sup>13</sup> Se refiere a *Signal-to-noise ratio*.

<sup>14</sup> Ser refiere a *Signal-to-noise and distortion*.

Las ecuaciones para cada medida son las siguientes:

$$THD = 20 \log \left( \frac{S}{D} \right)$$

donde

$$S = \text{Valor rms de la fundamental} \quad (10)$$

$$D = \sqrt{(A_1^2 + A_2^2 + \dots + A_n^2)}$$

$A_n$  = Valor rms del armónico número n.

$$SNR = 20 \log \left( \frac{S}{N} \right)$$

donde

$$S = \text{Valor rms de la fundamental} \quad (11)$$

$$N = \sqrt{(N_1^2 + N_2^2 + \dots + N_n^2)}$$

$N_n$  = Valor rms de la componente de ruido número n.

$$SINAD = 20 \log \left( \frac{S}{N + D} \right)$$

donde

$S$  = Valor rms de la fundamental

$$D = \sqrt{(N_1^2 + N_2^2 + \dots + N_n^2 + A_1^2 + A_2^2 + \dots + A_n^2)}$$

$N_n$  = Valor rms de la componente de ruido número n.

$A_n$  = Valor rms del armónico número n.

### 3.4 Cálculo de la varianza

Parte del instrumento ADDA incluye el cálculo de la varianza de las dos señales a la salida del filtrado adaptable. Es necesario el cálculo de la varianza para estimar la potencia de las dos señales y lograr expresar la distorsión y el ruido de la señal de prueba, como una medida en decibelios.

El cálculo de la varianza no es trivial. Se van a mostrar los tres métodos más conocidos para calcular la varianza, todos ellos entregan el mismo resultado en aritmética exacta, pero al ser trabajados en un procesador pueden dar diferentes resultados.

#### *Fórmula directa*

Matemáticamente el cálculo de la varianza puede ser definido de la siguiente manera:

$$s^2 = \frac{1}{n-1} \sum (x_i - \bar{x})^2 \quad (12)$$

Para calcular la varianza con esta fórmula se requiere almacenar un registro con el número ( $n$ ) de muestras que se van a analizar. Asimismo, se deben hacer dos barridos de las muestras del registro: El primero para hallar la media ( $\bar{x}$ ), y el segundo para hallar la varianza ( $s^2$ ).

Este método fue el ideado inicialmente para hallar la varianza en el diseño de ADDA.

### ***Suma de cuadrados***

Una manera diferente de hallar la varianza es la siguiente:

$$s^2 = \frac{1}{n(n-1)} \left( n \sum_{i=1}^n x_i^2 - \left( \sum_{i=1}^n x_i \right)^2 \right) \quad (13)$$

A primera vista esta fórmula es más efectiva en términos de recursos, comparada con la primera, ya que puede encontrar la varianza haciendo un solo barrido en el registro. Pero después de una mejor inspección resulta siendo poco viable en su implementación en el procesador

Si los valores de  $x$  son muy grandes y la diferencia entre los dos operandos es pequeña, la evaluación de la ecuación requeriría encontrar un número pequeño como la resta entre dos números muy grandes causando así problemas de cómputo[15] [16] .

Lo anterior genera una pérdida de precisión tan crítica, que en algunos casos la ecuación entrega un valor de varianza negativo, cuando la varianza siempre debe ser positiva.

### ***Método de Welford (Varianza continua)[14]***

El último método para calcular la varianza fue expuesto por B. P. Welford en 1962. Se podría decir que es el método más preciso de todos, además siempre entrega resultados mayores que cero.

Este algoritmo calcula una varianza continua, lo cual significa que no es necesario utilizar registros; sólo se necesita una ventana inicial de valores y a medida que entra un nuevo valor de  $x$  se puede calcular la varianza.

La varianza continua podría verse como una ventana corrediza que va entregando el valor de la varianza en tiempo real.

El algoritmo es el siguiente:

Inicializar  $M_1 = x_1$  y  $S_1 = 0$ .

Para las siguientes  $x$ 's, utilizar las siguientes fórmulas:

$$M_k = M_{k-1} + (x_k - M_{k-1})/k$$

$$S_k = S_{k-1} + (x_k - M_{k-1}) * (x_k - M_k).$$

Para  $2 \leq k \leq n$ , El  $k$ -ésimo estimado de la varianza es  $s^2 = S_k / (k - 1)$ .

Para el diseño de ADDA se hizo uso del método de fórmula directa y el de la varianza continua.

## 4 ESPECIFICACIONES

El Instrumento ADDA puede ser representado por el siguiente Diagrama General:



Figura 8: Diagrama General del funcionamiento Instrumento ADDA

**Las características más importantes del instrumento ADDA son las siguientes:**

- Entrega de Medida de SINAD con un Rango: 0-78 dB
- Puede medir la distorsión de señales con su frecuencia más alta en 30 kHz.
- Variaciones en el valor de SINAD<sup>15</sup> de 0,1%
- Entrega la medida del Porcentaje de Ruido y Distorsión de la señal de prueba
- Muestra las señales a la salida del filtro adaptable en tiempo real.

**Condiciones óptimas del instrumento ADDA:**

- Amplitud Máxima Señal de entrada: 2.82 Vp
- Comportamiento en Frecuencia:

Entrega de una medida acertada con una frecuencia de muestreo ( $f_m$ ) de por lo menos 3.2 veces el valor de la frecuencia más alta de la señal de prueba.

- Valor parámetro de tamaño de paso:  $\mu = 1E-11$
- Valor número de pesos del filtro: N=30

En la figura 9 se puede apreciar apreciar los bloques constitutivos del instrumento ADDA.

<sup>15</sup> En condiciones óptimas.

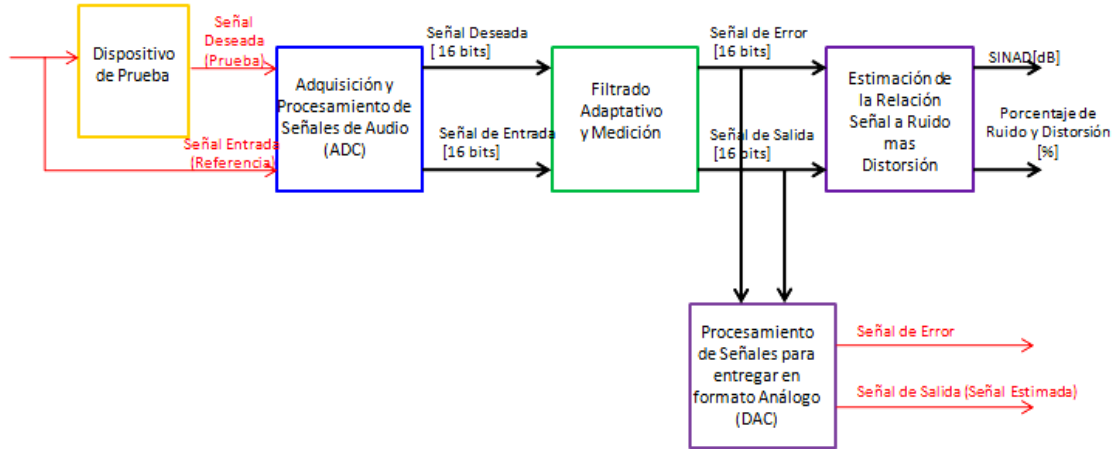


Figura 9: Diagrama de Bloques Constitutivos Instrumento ADDA

## 5 DESARROLLOS

### 5.1 Simulaciones

El Instrumento ADDA primero fue ideado y trabajado en simulaciones en MATLAB, Simulink. Todas las simulaciones en Simulink del instrumento se realizaron en paralelo con un estimador de THD<sup>16</sup>(Distorsión Armónica Total) que a diferencia del Instrumento ADDA necesitaba como parámetros de entrada la frecuencia de muestreo de la señal digitalizada ( $f_s$ ) y la frecuencia fundamental ( $f_i$ ). Esto se utilizó para validar las medidas del instrumento.

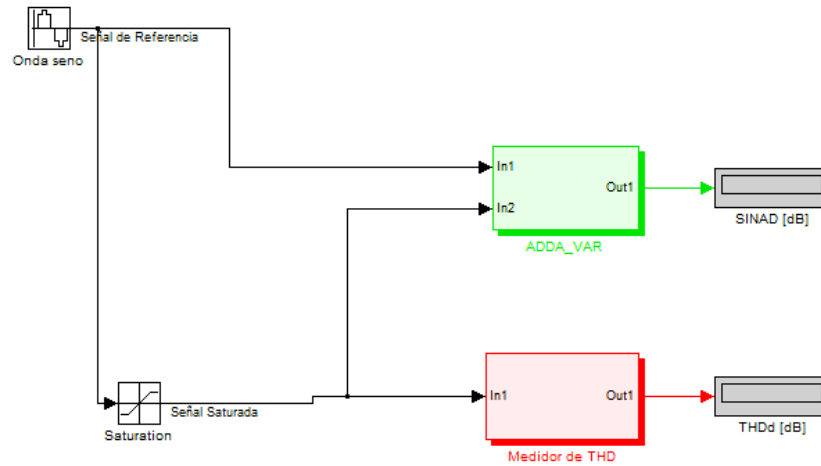


Figura 10: Simulación Instrumento ADDA EN MATLAB en paralelo con el Medidor de THD

El instrumento ADDA fue implementado de dos maneras en Simulink. Una utilizando la Varianza con registros y otra utilizando la Varianza continua (*Running Variance*).

---

<sup>16</sup> Total Harmonic Distortion



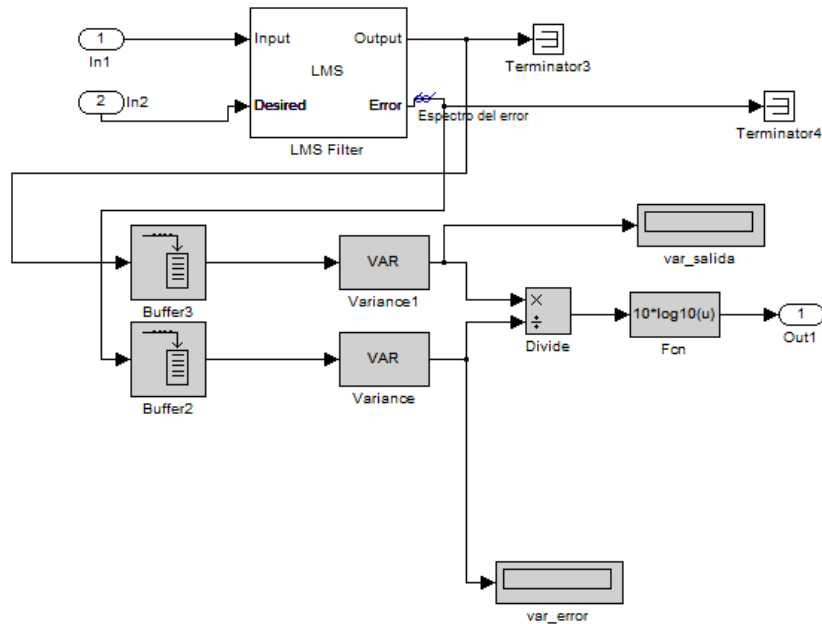


Figura 11: Simulación Instrumento ADDA utilizando *Variance*

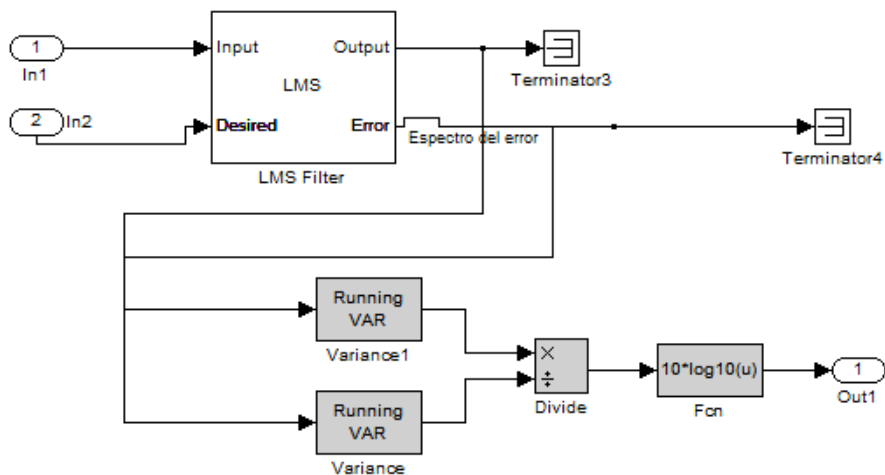


Figura 12: Simulación Instrumento ADDA utilizando *Running Variance*

Como se puede apreciar en las Figuras 11 y 12, las implementaciones son iguales excepto en el cálculo de la varianza.

## 5.2 Implementación en hardware

El método para la implementación en hardware jugó un gran papel en la investigación previa a la aplicación del instrumento, debido a que el hardware puede limitar o favorecer el diseño. Para la implementación del instrumento se tuvieron principalmente dos opciones y estas fueron una FPGA (*Field Programmable Gate Array*) y un DSP (*Digital Signal Processor*).

Basándose en la disponibilidad en el Laboratorio de Electrónica de la universidad se realizó una comparación entre las dos tarjetas de desarrollo: *Stratix II DSP Development Board* y *TMS320C6713 DSK*.

Característica	DSP ( <i>TMS320C6713 DSK</i> )	FPGA ( <i>Stratix II DSP Development Board</i> )
Procesamiento de señales de entrada	<p>Códec diseñado para manejar especialmente señales de audio y el ADC puede entregar señales hasta de 16 bits por cada canal.</p> <p>Usa el: tlv320aic23</p>	<p>El ADC entrega máximo señales de 12 bits</p> <p>Además, si se requiere un filtrado anti-solapamiento en la etapa de entrada, éste debe ser instalado externamente (no incluido en la tarjeta) [17]</p> <p>Usa el: AD9433</p>
Uso de notación punto flotante	<p>El procesador TMS320C6713 es un procesador de punto flotante. Esto es favorable, pues con el mismo número de bits se puede lograr mayor resolución. Además, para la implementación del filtrado adaptable, el uso de notación de punto flotante es la mejor metodología de diseño a usar: las simulaciones de estos algoritmos se hacen en punto flotante; si se decidiera utilizar punto fijo se deberían incluir constantes [18] y elaborar conversiones que no sólo agregarían complejidad, sino que afectarían el comportamiento de estos algoritmos.</p>	<p>Trabajar punto flotante en la FGA es posible pero complejo debido a que usualmente se logra incluyendo una unidad de punto flotante hecha a la medida. Este tipo de unidades utilizan bastantes recursos lógicos, y debido a esto, sólo una pequeña cantidad de unidades de punto flotante pueden ser usadas en todo el diseño y deben ser compartidas entre procesos. Esto último no permite aprovechar completamente la ventaja de <i>pipeline</i><sup>17</sup> que es posible con las FPGAs, siendo así un método bastante ineficiente. [18]</p>

Tabla 4. Tabla comparativa de FPGA vs DSP (*Stratix II DSP Development Board* y *TMS320C6713 DSK*).

El instrumento fue implementado usando el kit de desarrollo “*TMS320C6713 DSP Starter Kit (DSK)*” desarrollado en conjunto por las empresas Texas Instruments y Spectrum Digital.

<sup>17</sup> Este término se interpreta como la segmentación de alguna tarea específica para hacer que su desarrollo sea más rápido y efectivo, manejando varios procesos de la misma tarea en paralelo.

Para programar este dispositivo se hizo uso del *software* que viene con el kit de desarrollo y este es: Code Composer Studio DSK versión 3.1 IDE. Con este *software* se puede programar en C/C++ y *assembly*.

### 5.3 Algoritmos de implementación

Antes de desarrollar los Algoritmos para la Implementación, se describió un diagrama general de operaciones del instrumento ADDA:

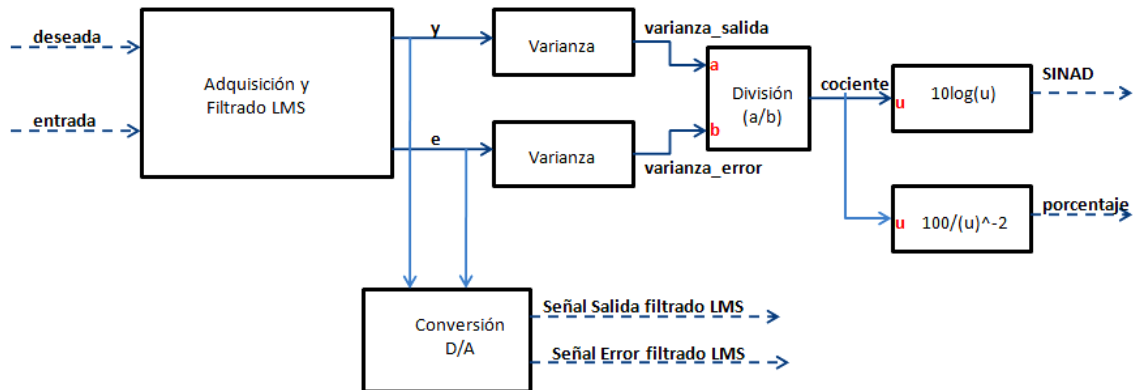


Figura 13: Diagrama General de Operaciones Instrumento ADDA

La adquisición de datos y conversión A/D se hizo con ayuda de unos archivos de soporte expuestos en el texto[10] . Estos archivos definen la suma de los dos canales, derecho e izquierdo, como una variable de tipo Uint32 (entero sin signo de 32 bits), tomando cada canal como un entero sin signo de 16 bits. Los archivos de soporte fueron usados gracias a su conveniencia en parametrización de valores y claridad para el manejo de los canales, estos archivos se encuentran en los Anexos Electrónicos.<sup>18</sup>Entre los archivos de soporte también se encuentra un archivo para soporte de las interrupciones (vectors\_intr.asm).

Para implementar el instrumento ADDA en el DSP se hizo uso de dos algoritmos. Los dos cumplen las mismas funciones, pero se diferencian en el método para el cálculo de la varianza de las señales a la salida del filtrado adaptable. El primer algoritmo se denomina algoritmo VAR y se basa en el método de cálculo de la varianza que hace uso de la fórmula Directa y el segundo se llama algoritmo runVAR y se basa en el método de Welford o de varianza continua. Para facilitar el entendimiento de la dinámica de los algoritmos, se muestran diagramas de flujo explicativos para cada implementación:

<sup>18</sup> Los nombres de los archivos de soporte son: c6713dskinit.c, c6713dsk.cmd, vectors\_intr.asm

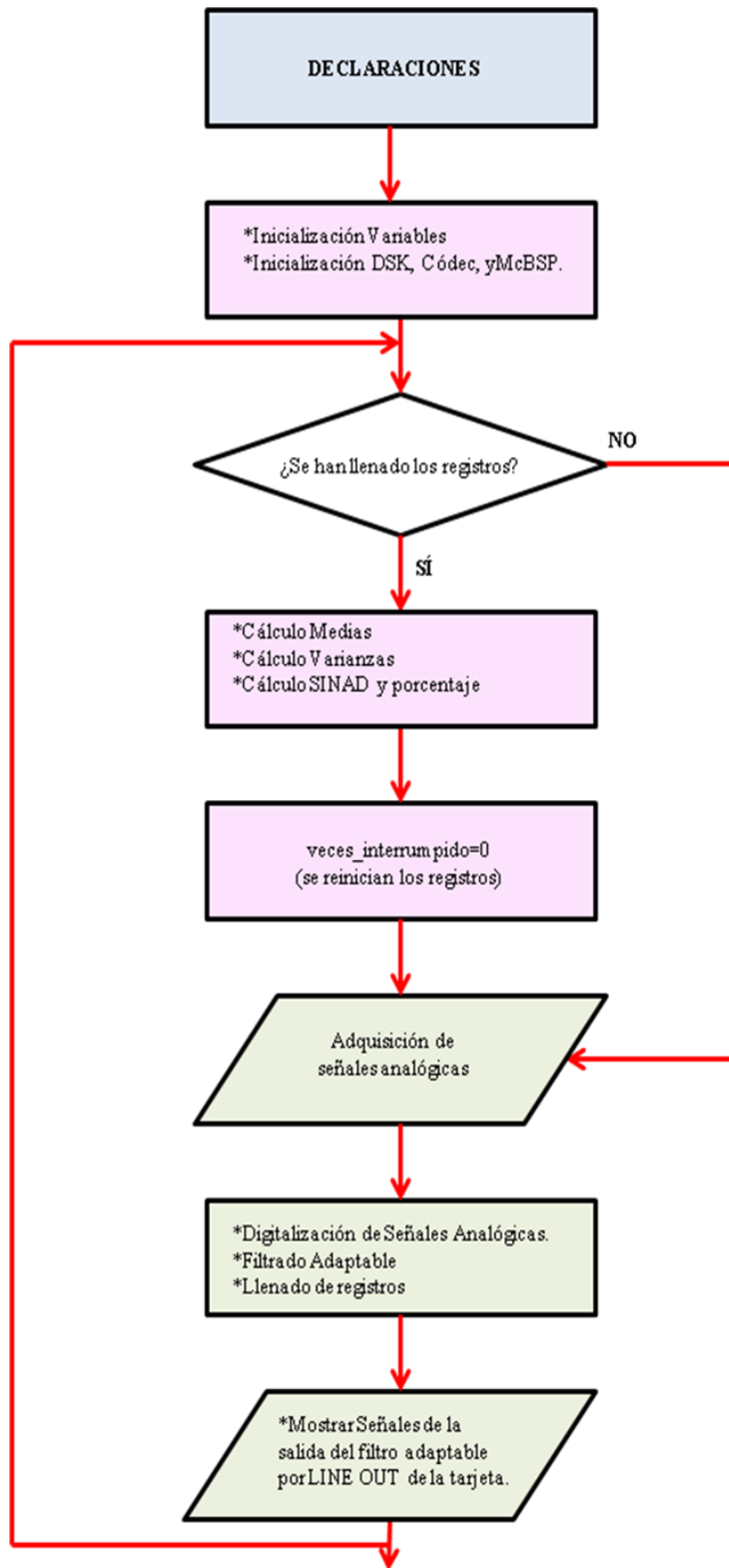


Figura 14: Diagrama de flujo algoritmo VAR

**El algoritmo VAR** se desarrolla parcialmente en tiempo real. Se están tomando valores de señales externas en tiempo real, pero el cálculo de la varianza, para hallar el valor de SINAD, se hace con los valores de las señales almacenados en registros. Para el caso del instrumento ADDA los registros son de 2048 muestras, pero este valor puede hacerse parametrizable.

A la primera instancia que se entra en el Algoritmo VAR es a las declaraciones; en este campo están las declaraciones de librerías de soporte, definiciones, macros, variables globales y la estructura para el códec (esta estructura se define basándose en los archivos de soporte utilizados para el códec).

Después de esto, se entra a la función *Main*<sup>19</sup> del programa, en el diagrama se ilustran los procesos que se realizan en la función *Main* con color rosado. Una vez se entra al *Main*, se inicializan algunas variables, y se inicializa el DSK, el códec y el McBSP<sup>20</sup>(con ayuda de los archivos de soporte).

Luego se pregunta si se han llenado los registros, que en este caso son de 2048 muestras. Si se han llenado los registros, se calculan las medias, las varianzas, el valor de SINAD y el porcentaje. Para el cálculo de SINAD se tiene la condición que el denominador del cociente de SINAD sea diferente de cero.

Acto seguido, se inicializa el índice “veces\_interrumpido”. En caso que no se hallan llenado los registros, el programa se dirige hacia la rutina de interrupción, que está compuesta por los bloques que se tienen con color verde.

En la rutina de interrupción se reciben los datos externos (adquisición de señales analógicas), se digitalizan, se filtran y se procede a llenar los registros, que van a ser usados para los cálculos en la función del *Main*. Asimismo, se muestran las señales a la salida del filtro, por medio del *LINE OUT*<sup>21</sup> de la tarjeta.

Para salir del loop infinito del algoritmo VAR se debe utilizar la instrucción *halt*<sup>22</sup>.

---

<sup>19</sup> Función principal del programa.

<sup>20</sup> *Multichannel Registered Serial Port*

<sup>21</sup> Línea de Salida

<sup>22</sup> interrumpir

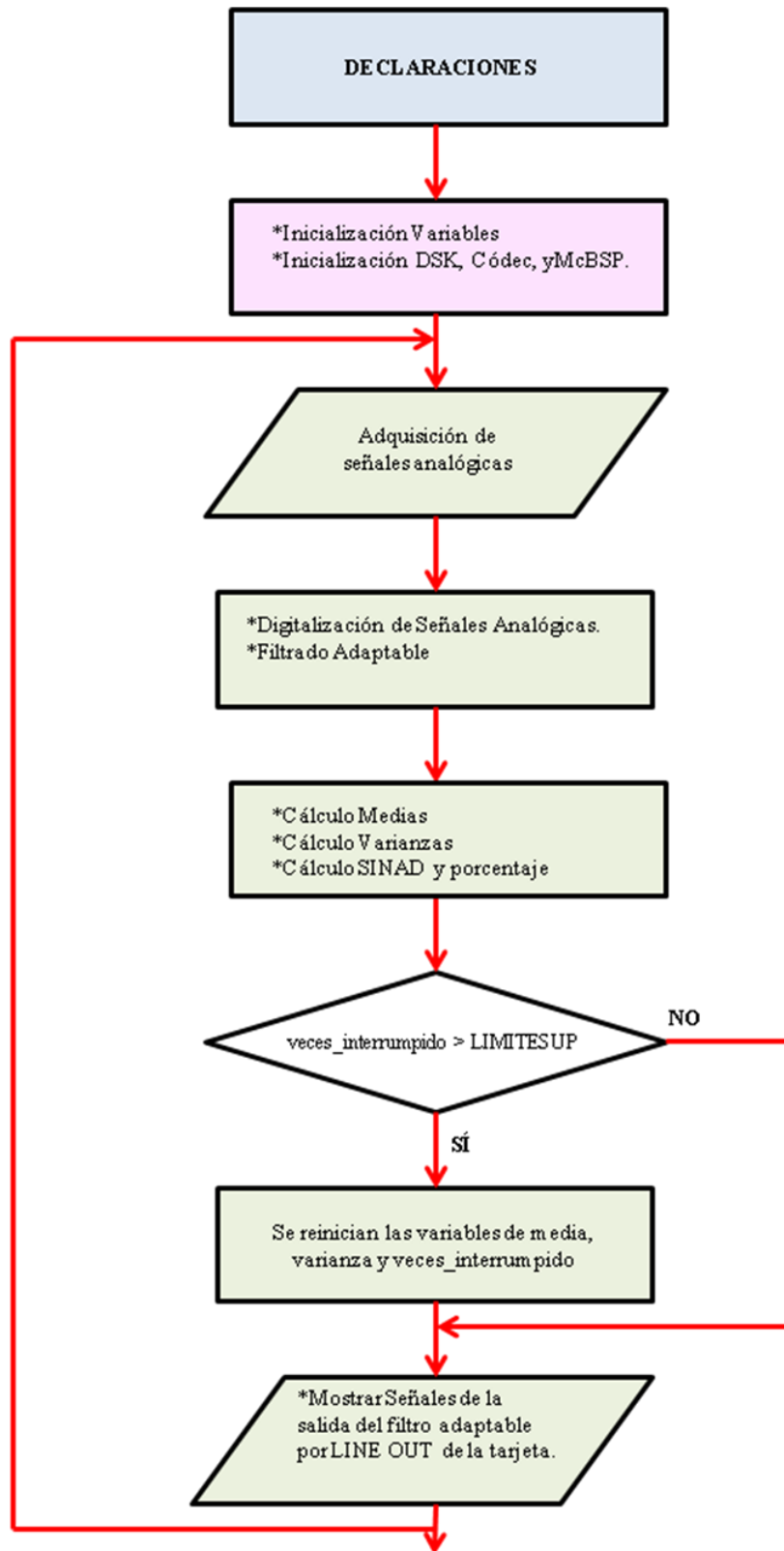


Figura 15: Diagrama de flujo algoritmo **runVAR**

**El algoritmo runVAR** de la misma manera que el algoritmo VAR, inicia en las Declaraciones, luego se entra a la función del *Main*, donde se inicializan variables, DSK, Códec y McBSP.

Acto seguido, se entra a la interrupción y se reciben las señales analógicas externas. Después de esto, se digitalizan las señales de entrada, se filtran.

En los cálculos siguientes existen algunas condiciones que no se incluyeron en el diagrama, porque no son condiciones que impidan entrar a la interrupción. Para calcular las medias y varianzas se debe tener como condición que el índice “veces\_interrumpido>1”, esta condición asegura que el denominador de una operación para hallar la varianza no tome el valor de cero (recordar la condición para hallar la varianza en el método de Welford). En el cálculo del SINAD se hace la misma validación que en el programa anterior, que el denominador del cociente del SINAD no tome el valor de cero; cuando esto suceda, se entrega un valor de SINAD igual a cero.

Luego de realizar los cálculos respectivos se presenta la condición que si el índice “veces\_interrumpido>LIMITESUP” se reinicien las variables de media, varianza y “veces\_interrumpido”. Esto se realiza para evitar que valores iniciales a la salida del algoritmo, influyeran en la medida de SINAD ya que inicialmente el valor de error del algoritmo no es mínimo.

Al final de la rutina de interrupción se muestran las señales de error y salida del filtrado adaptable haciendo uso del DAC del códec para poder apreciar las señales en un osciloscopio.

Para salir del loop infinito del algoritmo runVAR se debe utilizar la instrucción *halt*.

Para mayor información acerca de los algoritmos usados, por favor remitirse a los Anexos Electrónicos<sup>23</sup>.

---

<sup>23</sup> El código *Source* del Algoritmo VAR es ADDA\_VAR.c y el código *Source* del Algoritmo runVAR es ADDA\_runVAR.c

## 6 ANÁLISIS DE RESULTADOS

### 6.1 Análisis piso de ruido

Se tomaron muestras entrando señales analógicas a la tarjeta y extrayendo las muestras digitalizadas para entender el comportamiento del códec.

Una de esas pruebas se realizó colocando los dos canales de entrada a cero voltios, para ver el piso de ruido del códec. Las gráficas resultantes son las siguientes:

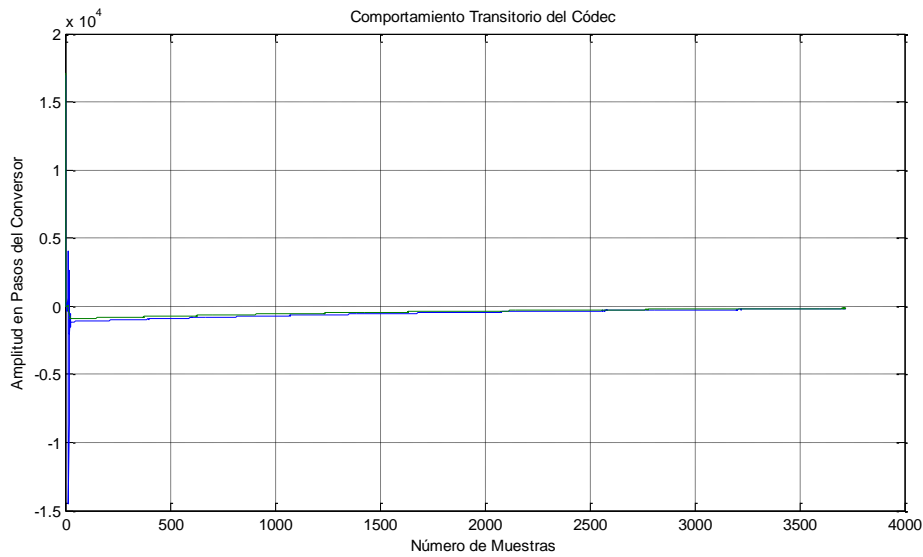


Figura 16: Comportamiento Transitorio del Códec

En la figura 16 se puede observar que hay un pico al inicio y luego un transitorio.

Haciendo zoom en la gráfica se puede observar mejor el pico:

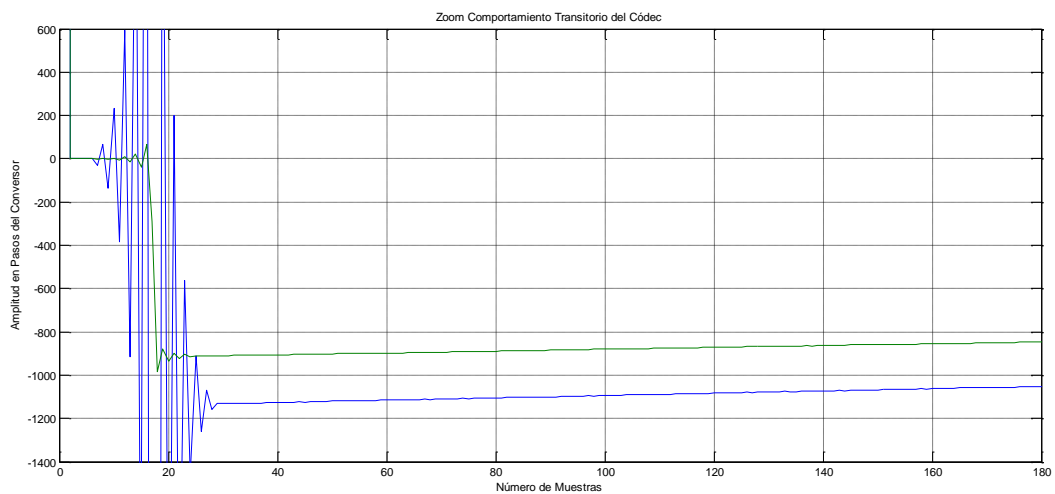


Figura 17: Zoom Comportamiento Transitorio del Códec



La señal del canal izquierdo es la señal en azul y la del canal derecho es señal en verde.

Después que pasa el pico la señal toma valores menores que cero y al pasar el tiempo se intenta estabilizar en cero.

Notando este comportamiento, se incluyó un retardo entre la adquisición del códec y la toma de muestras y se pudo observar el piso de ruido en estado estable:

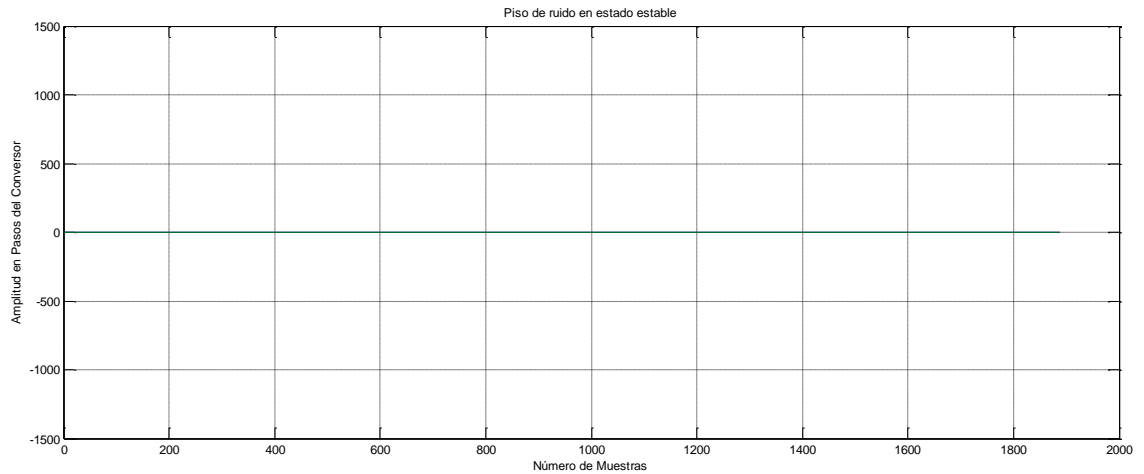


Figura 18: Piso de ruido en estado estable

Haciendo zoom en la gráfica se puede observar:

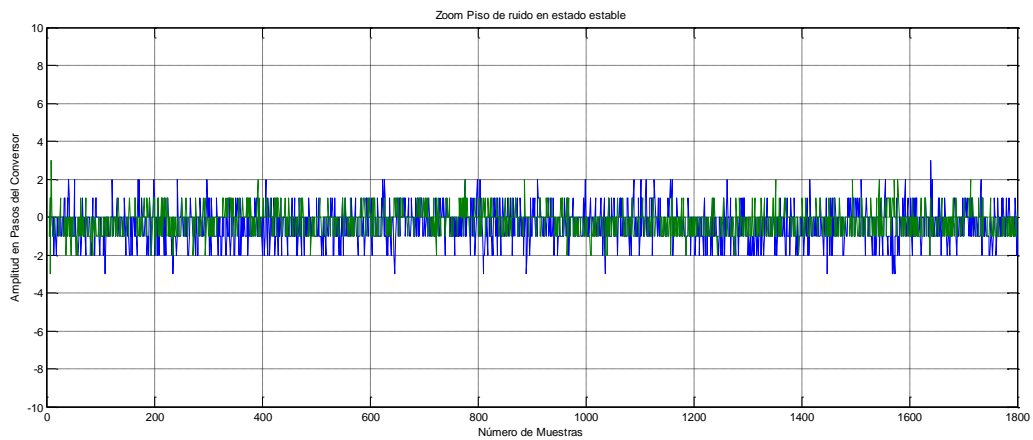


Figura 19: Zoom Piso de ruido en estado estable

A pesar que existen variaciones, se dan en valores bastante pequeños.

El efecto del pico al inicio de la conversión del códec y el comportamiento fuera de estado estable puede observarse en las siguientes señales digitalizadas y luego observadas en MATLAB:

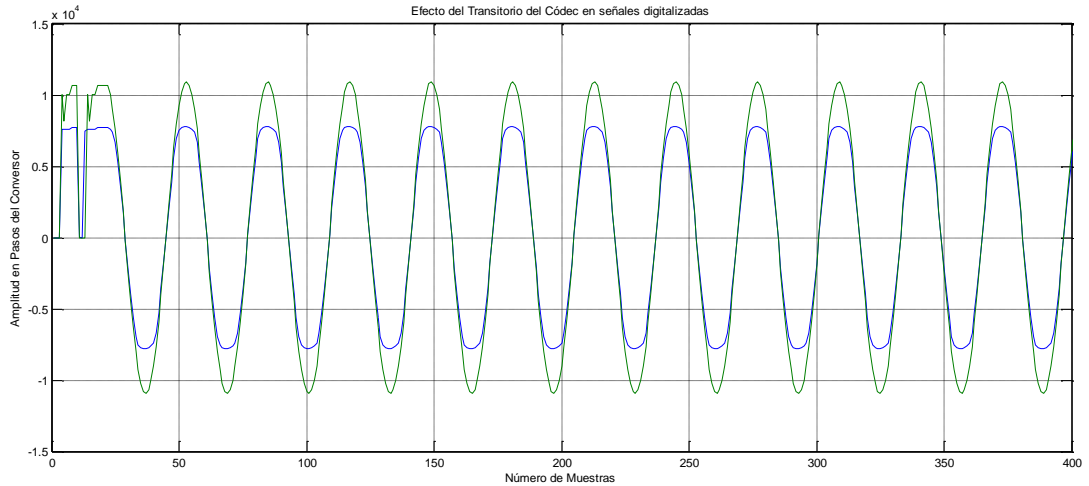


Figura 20: Efecto del Transitorio del Códec en señales digitalizadas

Como se puede observar existe una alteración de la señales al inicio de la conversión y luego el comportamiento del Códec se estabiliza y digitaliza las señales de modo correcto.

## 6.2 Variación frecuencia señal de entrada (fi)

### *Especificaciones de la prueba*

La señal de entrada es saturada utilizando dos diodos en anti-paralelo. El montaje del circuito de prueba es el siguiente:

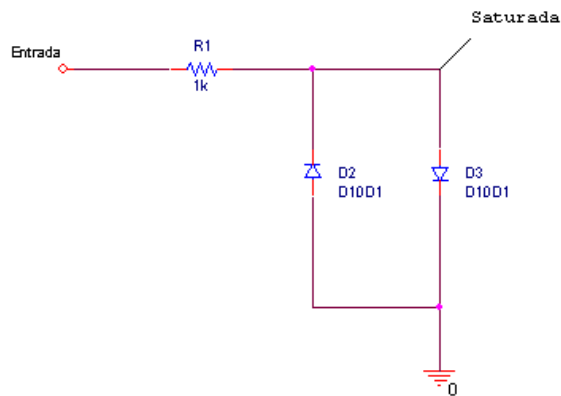


Figura 21: Circuito para saturar la Señal de Entrada

## Resultados

Se tomaron medidas con el Instrumento ADDA, variando el factor  $u$ , o parámetro de tamaño de paso. Asimismo, se tomaron medidas con Analizadores de Distorsión Analógicos

Frecuencia $f_i$	VAR ( $u=1E-10$ )	VAR ( $u=1E-11$ )	runVAR ( $u=1E-10$ )	runVAR ( $u=1E-11$ )	Instrumento Analógico <sup>24</sup>
200Hz	36,29912dB 1,531243%	33,01756dB 2,2342%	36,356968dB 1,521078%	33,049294dB 2,226052%	1,29%
500Hz	33,12045dB 2,207891%	32,80676dB 2,289086%	32,95013dB 2,251611%	32,870529dB 2,272341%	2,27%
1kHz	32,5999dB 2,344257%	31,71147dB 2,596728%	32,59124dB 2,346594%	32,000751dB 2,511670%	2,51%
5kHz	35,48822dB 1,681082%	31,54339dB 2,647466%	34,93039dB 1,792589%	33,329063dB 2,155494%	2,65%
10kHz	51,23171dB 0,2744191%	40,24777dB 0,9718778%	49,39819dB 0,3389147%	40,93002dB 0,8985174%	2,62%
12kHz	58,21704dB 0,1227857%	54,76041dB 0,1828014%	52,53146dB 0,2362801%	47,43393dB 0,4249165%	2,61%
15kHz	42,29258dB 0,76801775%	38,04799dB 1,251989%	46,08936dB 0,4960894%	36,51705dB 1,493301%	2,62%

Tabla 5: Valores de SINAD para diferentes frecuencias de la Señal de Entrada ( $f_i$ ) al Instrumento ADDA con  $f_s=32$ kHz

Como se puede ver, las medidas del instrumento ADDA dejan de ser confiables cuando  $f_i=10$ kHz. Así pues sería recomendable utilizar una frecuencia de muestreo de por lo menos 3,2 veces el valor de la frecuencia de la señal de entrada, para seguir recibiendo medidas confiables del instrumento ADDA.

Cabe aclarar que las medidas del instrumento Analógico se mantuvieron relativamente estables en comparación con las del instrumento ADDA, porque para cada medida se debía cambiar el rango de frecuencia en el instrumento y calibrar el instrumento como mínimo.

### 6.3 Variación frecuencia de muestreo ( $f_s$ )

#### *Especificaciones de la prueba*

En esta prueba se tomaron medidas variando la frecuencia de muestreo del instrumento de 8 kHz hasta 96 kHz. Las señales de prueba tenían las siguientes especificaciones:

$f_i = 1$  kHz

Señal saturada en el 90% del nivel, para los niveles inferior y superior.

El filtro LMS tenía las siguientes especificaciones:

$N=30$

<sup>24</sup> Para las medidas del “Instrumento Analógico” se tuvieron en cuenta las medidas hecho con los instrumento: HAMEG HM 8027 y LEADER LDM 171, y se utilizó una sola cifra porque los resultados eran bastante similares.

$$u=1E-12$$

Tamaño del registro<sup>25</sup> (para la medida de SINAD)= 2048.

Para las simulaciones en Simulink se utilizaron las mismas especificaciones a diferencia de las siguientes del filtro LMS:

$$N=4$$

$$u=0.1$$

Estos valores son radicalmente diferentes a los usados en el instrumento, porque las señales usadas en los dos métodos son diferentes en su naturaleza (esto se va a analizar más adelante.)

### Resultados

Cada uno de los valores de las medidas en las tablas comparativas fue tomado como el promedio de 150 muestras. Las tablas que contienen todos estos datos se encuentran en los Anexos Electrónicos<sup>26</sup>. Los únicos valores de las tablas que no fueron encontrados como el promedio de un grupo de muestras, son los que tienen asterisco y los valores de simulación en MATLAB, que fueron entregados como el valor de estabilización.

Frecuencia de muestreo (fs)	Señales Internas				
	Algoritmo VAR	Algoritmo RUNVAR	Algoritmo VAR Simulink	Algoritmo RUNVAR Simulink	Medidor THD Simulink
8 kHz	25,240456	26,021164	21,8	22,98	25,58
16 kHz	27,240243	27,586012	26,97	25,34	27,96
32 kHz	27,122748	27,373712	27,1	26,15	27,64

<sup>25</sup> Para el algoritmo VAR este es el tamaño del registro para tomar cada medida, pero para el algoritmo runVAR este es el tamaño de la ventana (las medidas se toman cada vez que entre una nueva muestra).

<sup>26</sup> Los archivos que contienen estas pruebas son los siguientes: VARIANDO\_fs8k.xls, VARIANDO\_fs16k.xls, VARIANDO\_fs32k.xls, VARIANDO\_fs48k.xls, VARIANDO\_fs96k.xls.

Frecuencia de muestreo (fs)	Señales Internas				
	Algoritmo VAR	Algoritmo RUNVAR	Algoritmo VAR Simulink	Algoritmo RUNVAR Simulink	Medidor THD Simulink
48 kHz	27,354757*	27,418768	28,01	27,18	27,56
96 kHz	27,535468*	27,412945	31,17	30,25	27,48

Tabla 6: Valores de SINAD para diferentes frecuencias de muestreo (fs) utilizando señales internas.

Esta primera tabla comparativa tiene valores generados con señales internas o que no deben ser digitalizadas. Las señales internas usadas para las pruebas de *Code Composer Studio* fueron generadas en el mismo programa *source*<sup>27</sup>, pero son usadas en tiempo real.

Como se puede observar, los valores de SINAD que entrega el instrumento ADDA no varían mucho a medida que cambia la frecuencia de muestreo. Esto se debe al hecho que no existe conversión A/D y el algoritmo LMS se adapta de muy buena manera.

Por otro lado, las medidas de SINAD entregadas por la simulación en Simulink variaron aumentando a medida que aumentaba la frecuencia de Muestreo.

Las medidas del medidor THD son relativamente estables porque reciben como parámetros los valores de la frecuencia de la señal de entrada (fi) y la frecuencia de muestreo (fs).

Frecuencia de muestreo (fs)	Señales Externas	
	Algoritmo VAR	Algoritmo RUNVAR
8 kHz	29,208050	29,648756

<sup>27</sup> Programa fuente o principal

Frecuencia de muestreo (fs)	Señales Externas	
	Algoritmo VAR	Algoritmo RUNVAR
16 kHz	27,455174	27,621423
32 kHz	27,142016	27,529586
48 kHz	26,382274	26,302225*
96 kHz	26,451855	27,358373

Tabla 7: Valores de SINAD para diferentes frecuencias de Muestreo (fs) utilizando señales externas.

Las pruebas para las señales externas fueron realizadas con unas señales analógicas generadas por el programa Goldwave[8] . Este programa crea las señales con una frecuencia de muestreo de 44100 Hz, que es una frecuencia estándar en audio, luego hace una conversión D/A y entrega las señales en formato analógico haciendo uso de la tarjeta de audio del computador.

El valor de SINAD, entregado por el instrumento ADDA analizando señales externas, disminuye a medida que aumenta la frecuencia de muestreo, comportándose de modo opuesto a las simulaciones en Simulink.

#### 6.4 Variación número de pesos del filtro (N)

##### *Especificaciones de la prueba*

Para esta prueba se utilizaron Señales Externas generadas por el programa Goldwave[8] , y Señales Internas generada al interior de Code Composer Studio.

Las Señales Externas tenían las siguientes especificaciones:

$$f_i = 1 \text{ kHz}$$

Señal Saturada en el 90% del nivel, para los niveles inferior y superior. De 1 Vp

El filtro LMS tenía las siguientes especificaciones:

$N=30$

$u=1E-12$

Tamaño del registro (para la medida de SINAD)= 2048.

Para las señales dentro de Simulink se utilizó un  $u=0.1$

### Resultados

Número de Pesos (N)	Señales Externas		Señales Internas			
	Algoritmo	Algoritmo	Algoritmo	Algoritmo	Algoritmo	Algoritmo
	VAR	RUNVAR	VAR	RUNVAR	VAR Simulink	RUNVAR Simulink
1	25,519663	27,371018	27,470439	27,365766	27,35	26,03
4	27,131831	27,353094	27,061607	27,446219	27,1	26,15
10	27,515051	27,531203	26,035719	27,447169	27,82	27,13
20	27,473408	27,527766	26,918026	27,419622	30,57	29,42
30	27,394890	27,528518	27,109565	27,405420	31,92	30,42

Tabla 8: Valores de SINAD para diferentes Números de Pesos del filtro (N)

Las medidas del SINAD con el instrumento ADDA mantienen un valor más o menos constante con el cambio de N, tomando valores levemente mayores con un valor de N mayor. Por otro lado, los valores de la SINAD que arrojan las simulaciones aumentan en mayor cantidad al aumento del

número de pesos del filtro. Asimismo, cabe anotar que a medida que se aumenta el número de pesos del filtro, se debe disminuir el parámetro de tamaño de paso  $u$ .

Adicionalmente, se tomó la medida de distorsión de la señal Analógica de prueba con el Analizador de Distorsión HM 8027 y el analizador entregó la medida de SINAD=27.64 dB. Como se puede observar, la medida es está en el mismo rango de los valores de SINAD entregados por el Instrumento ADDA, corroborando la validez de las medidas.

## 6.5 Valores máximos

### *Especificaciones de la Prueba*

Para el caso de los valores máximos se tiene la misma señal como Entrada y Deseada al algoritmo.

Con esta medida se puede hacer una medida estimada los bits efectivos (ENOB) del instrumento, que estarían limitados por el códec de la tarjeta.

Utilizando la siguiente ecuación:[13]

$$ENOB = \frac{SINAD - 1,76dB}{6,02} \quad (14)$$

En la ecuación anterior, se asume una amplitud de escala completa, así que a la anterior ecuación se le debe incluir un factor de escala para los casos en los que no se tiene escala completa, generándose así la siguiente ecuación:

$$ENOB = \frac{SINAD_{MEDIDO} - 1,76dB + 20 \log \left( \frac{Amplitud \_ Escala \_ Completa}{Amplitud \_ de \_ Entrada} \right)}{6,02} \quad (15)$$

La amplitud de voltaje para escala completa (*fullscale*[24] ) para la entrada al códec es de 1Vrms, pero la tarjeta C6713 DSK tiene un divisor de voltaje con una ganancia de 0.5 para las entradas por LINE IN, entonces la amplitud de escala completa sería de 2 Vrm ó 2,8284 Vp.



## Resultados

Mirando las simulaciones en *Simulink*:

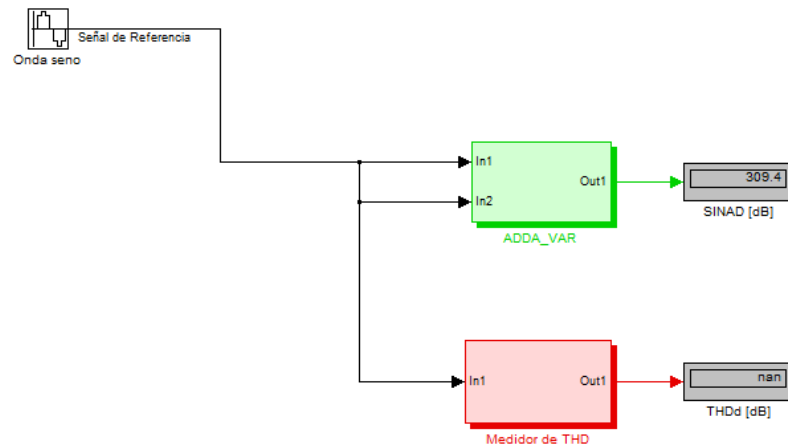


Figura 22: Simulación en MATLAB para encontrar los Valores Máximos del Instrumento ADDA

Utilizando la primera ecuación con el valor de SINAD darían 51,103 ENOB. Esto tiene sentido porque Matlab trabaja con resolución de 64 bits.

Luego se hicieron pruebas con el Instrumento ADDA, y se tomó el valor fijo de  $\mu=1E-10$  para todos los casos. Cada uno de los valores de las medidas en las tablas comparativas fue tomado como el promedio de 150 muestras. Las tablas que contienen todos estos datos se encuentran en los Anexos Electrónicos<sup>28</sup>.

---

<sup>28</sup> El archivo que contiene estas pruebas es el siguiente: VALOR\_MAX.xls

	Fuente Señales de Prueba					
	Generador de Audio Leader LAG 120		Generador de Ondas Arbitrarias Agilent 33120		Señales de entrada generadas al interior de Code Composer Studio	
	Algoritmo VAR	Algoritmo runVAR	Algoritmo VAR	Algoritmo runVAR	Algoritmo VAR	Algoritmo runVAR
<b>Amplitud Señal Entrada</b> [Vp]	2	2	2	2	1(float de 23405)	1(float de 23405)
<b>Promedio</b> [dB]	79,021086	79,081429	68,822243	78,929798	141,169940	134,446249
<b>Valor Máximo</b> [dB]	79,527794	79,508804	73,827469	79,488510	145,909409	134,853180
<b>Valor Mínimo</b> [dB]	78,550545	78,482635	40,167130	78,168839	136,914154	134,062912
<b>Porcentaje Variación del Valor.</b> [%]	0,24915745	0,257796052	10,5558413	0,318108254	1,319433088	0,115402808
<b>Estimado ENOB</b>	13,108	13,105	12,162	13,102	25,135	23,299

Tabla 9: Valores Máximos de SINAD utilizando diferentes Señales de Prueba

Para el cálculo de los ENOB se tomaron 3 cifras significativas.

Como se puede ver los valores de las señales analógicas arrojan valores de ENOB alrededor de 13, esto tiene sentido ya que la resolución del instrumento ADDA está limitada por la resolución del códec TLV320AIC23 que es de 16 bits para cada canal.

En cambio, para las señales internas el valor de ENOB es mucho mayor. Esto se debe a que la longitud de palabra que maneja el procesador es de 32bits.

El analizador de distorsión analógico LEADER LDM 171 arrojó valores máximos alrededor de 80dB, que equivaldría a 0.01% de distorsión de la señal de prueba.

## 6.6 Prueba integrada 1

### *Especificaciones de las pruebas integrada 1 y 2*

El propósito de esta prueba fue hacer un análisis integrado del funcionamiento del instrumento en paralelo con el funcionamiento del modelo del instrumento en Simulink.

La prueba consistió en los siguientes pasos:

1. Generar las señales analógicas para prueba (Deseada, Entrada).
2. Procesar las señales con la tarjeta y llenar unos registros con los valores de sus muestras digitalizadas (1 cada 1/fs).
3. Recolectar un número de las muestras de las dos señales, e importarlas al *workspace*<sup>29</sup> de MATLAB.
4. Modificar el formato de las señales para poder trabajarlas en Simulink como entradas a las simulaciones del instrumento ADDA para los algoritmos VAR y runVAR.
5. Tomar medidas del SINAD con los algoritmos VAR y runVAR implementados en la tarjeta y comparar con los valores de las simulaciones.

Para el primer paso se utilizó el programa Goldwave[8] y se generaron las señales deseada y entrada con las siguientes especificaciones:

*Señal saturada al 90% (esta sería la señal deseada)*

*Amplitud= 1 Vp*

*fi=1 kHz*

*fs(programa)=44100 Hz*

Las señales generadas por Goldwave se observaron en el osciloscopio:

---

<sup>29</sup> Espacio de trabajo.

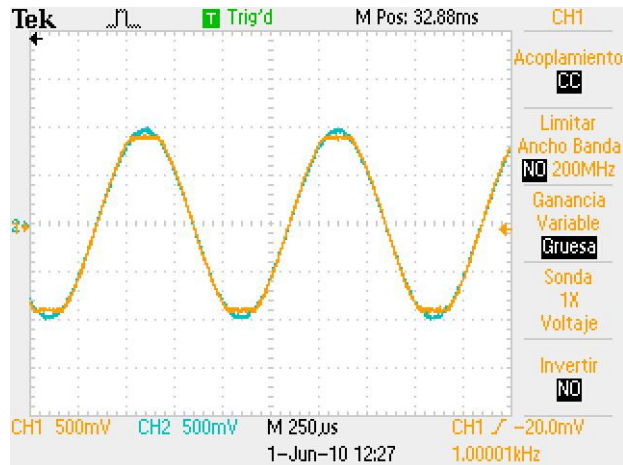


Figura 23: Señales de entrada Prueba Integrada 1

La señal en amarillo es el canal izquierdo (o la deseada)

La señal en azul es el canal derecho (o la entrada)

Luego de esto las señales fueron procesadas por el códec de la tarjeta con las siguientes especificaciones:

$$fs(\text{procesamiento en CCS})=32 \text{ kHz}$$

$$\text{Número de muestras tomadas}=2757$$

Las señales fueron almacenadas en un registro y luego exportadas a un archivo de texto utilizando la funciones *printf* (para imprimir las señales en un formato determinado) y *Save Output to File* (para guardar lo que se imprimió en un archivo de texto). El comando *printf* al darle formato a los datos los multiplica por  $1 \times 10^6$ , pero las señales a la salida del códec se ven atenuadas en  $1 \times 10^2$ ; por esto es que el valor más alto que se ve de las muestras es  $1 \times 10^4$ . De todos modos, como se quiso hacer un análisis de los datos reales, al importar las muestras a MATLAB se dividió el valor de cada una por  $1 \times 10^6$ .

Después de haber importado las señales al *workspace* así se pueden apreciar en MATLAB:

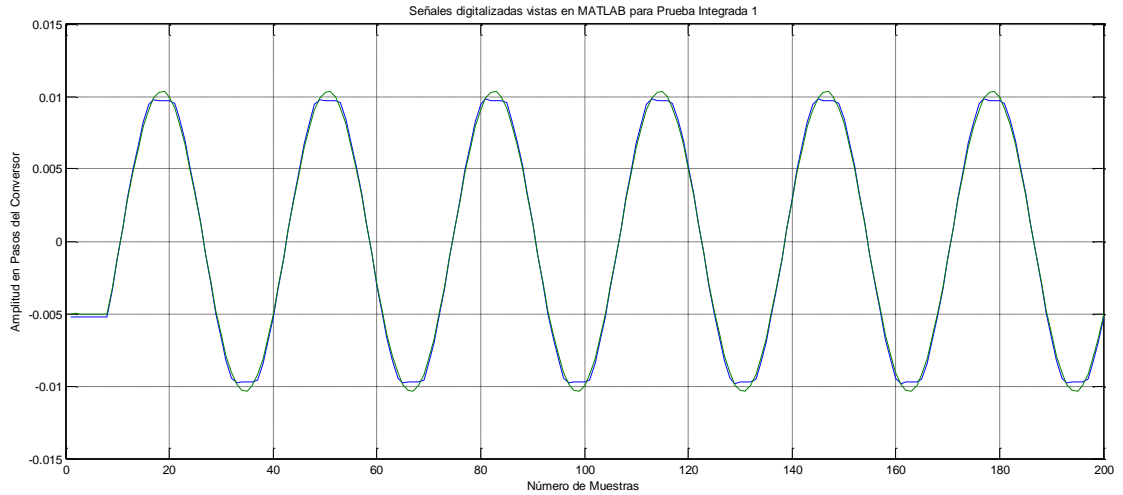


Figura 24: Señales digitalizadas vistas en MATLAB para Prueba Integrada 1

Estando la Señal de Entrada en color verde y la Señal Deseada en color azul.

Para tomar las señales como entradas de las simulaciones en Simulink se crearon estructuras para cada una, donde cada estructura contenía un parámetro de tiempo (vacío para poder utilizar la opción de repetición cíclica) y un parámetro de valores que contenía las muestras de la señal. Adicionalmente, se le indicaba a la simulación la frecuencia de muestreo ( $f_s$ ) en la que fueron digitalizadas las señales.

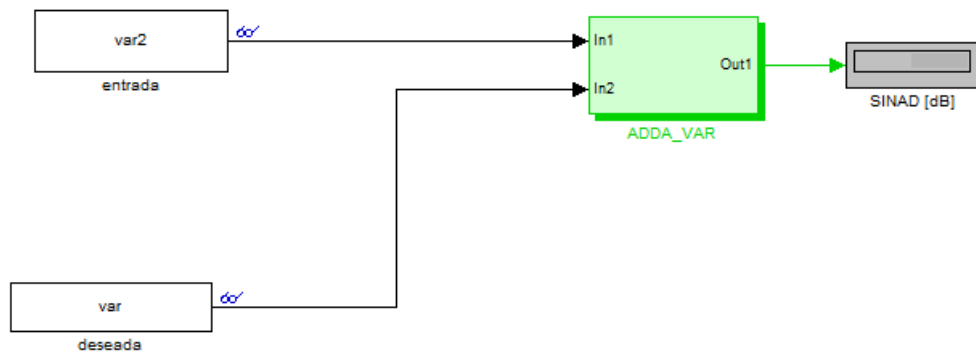


Figura 25: Simulación Prueba Integrada 1 utilizando señales externas

### ***Algoritmo VAR en Simulink usando***

$$u=4$$

$$N=4$$

Entregó un valor de SINAD=27,53 dB.

Las gráficas de las señales en tiempo se pueden apreciar a continuación:

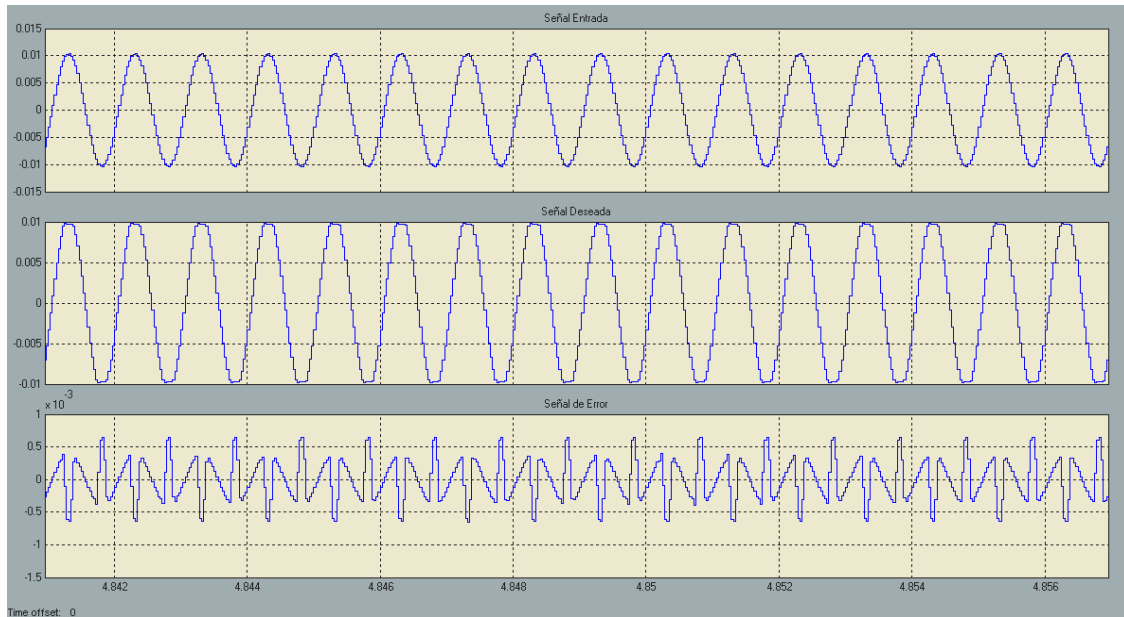


Figura 26: Señales en tiempo (Entrada, Deseada y Error) para Prueba Integrada 1 utilizando algoritmo VAR

Se puede apreciar cómo la Señal de Error tiene promedio cero (en cada ciclo) y cómo además no tiene componente fundamental. Adicionalmente, se observa la manera en la que el algoritmo se adapta ante los recortes de la Señal Deseada.

### ***Algoritmo runVAR en Simulink usando***

$$u=6$$

$$N=4$$

Entregó un valor de SINAD=26,73 dB

Las gráficas de las señales en tiempo se pueden apreciar a continuación:

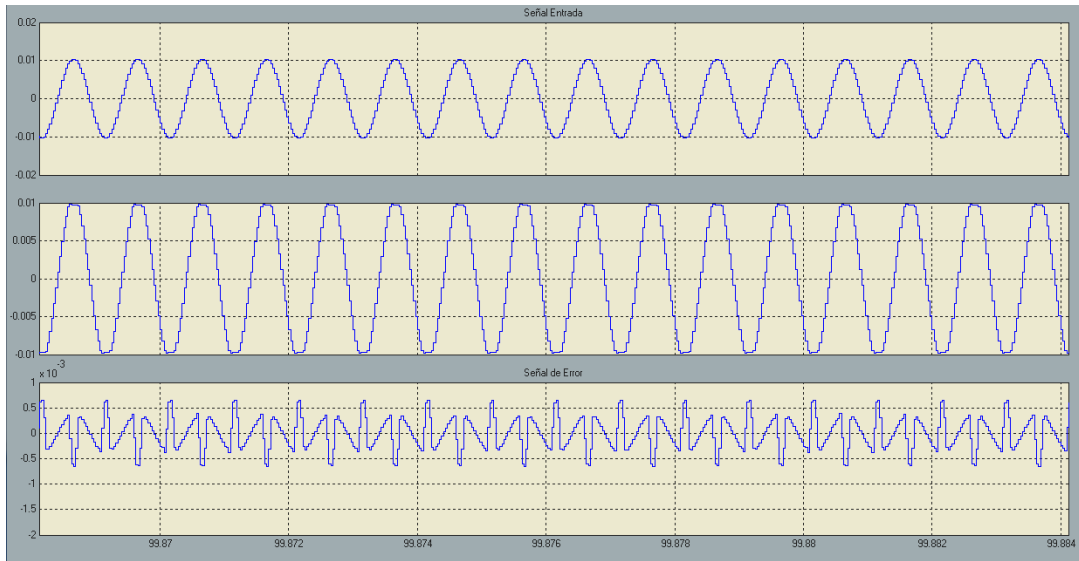


Figura 27: Señales en tiempo (Entrada, Deseada y Error) para Prueba Integrada 1 utilizando algoritmo runVAR

Observando las señales en tiempo se puede observar el mismo comportamiento que con el Algoritmo VAR, esto se debe a que el filtrado es el mismo, la diferencia entre estos algoritmos está dada en el método para calcular los valores de varianza de las señales a la salida del filtro.

El valor entregado por el instrumento HAMEG HM8027 fue de una distorsión de 4,64 % o de 26,67 dB de SINAD.

Las medidas entregadas por el Instrumento ADDA utilizando exactamente las mismas señales que fueron digitalizadas e importadas a MATLAB fueron las siguientes:

	Medidas SINAD utilizando el algoritmo VAR			
	u=1E-9	u=1E-10	u=1E-11	u=1E-12
<b>Promedio SINAD</b> [dB]	31,322592	27,106942	27,412869	27,355044
<b>Valor Máximo SINAD</b> [dB]	31,567287	27,133259	27,484127	27,539896
<b>Valor Mínimo SINAD</b> [dB]	27,772083	26,872950	27,149504	26,185324
<b>Porcentaje Variación SINAD</b> [%]	1,209009451	0,170894597	0,456772984	0,848157644
<b>Promedio Porcentaje SINAD</b> [%]	2,718471	4,412239	4,259922	4,289475

Medidas SINAD utilizando el algoritmo VAR				
	u=1E-9	u=1E-10	u=1E-11	u=1E-12
<b>Valor Máximo Porcentaje SINAD</b>				
[%]	4,086917	4,532653	4,390609	4,906071
<b>Valor Mínimo Porcentaje SINAD</b>				
[%]	2,640193	4,398829	4,224678	4,197640

Tabla 10: Medidas SINAD utilizando el algoritmo VAR para la Prueba Integral I

En estas pruebas también se quiso evaluar el comportamiento del algoritmo variando el parámetro de tamaño de paso  $u$ . El  $u$  de mayor valor con el que funcionó correctamente el instrumento ADDA fue de  $u=1E-9$ . Y el  $u$  óptimo para todas las medidas fue el de  $u=1E-10$ , ya que usando ese valor se vieron las variaciones más pequeñas del valor entregado de SINAD por el instrumento.

Medidas SINAD utilizando el algoritmo runVAR				
	u=1E-9	u=1E-10	u=1E-11	u=1E-12
<b>Promedio SINAD</b>				
[dB]	31,466455	27,123339	27,460013	27,497449
<b>Valor Máximo SINAD</b>				
[dB]	31,578848	27,135162	27,479044	27,511478
<b>Valor Mínimo SINAD</b>				
[dB]	31,283907	27,048042	27,413872	27,468691
<b>Porcentaje Variación SINAD</b>				
[%]	0,228594673	0,042006739	0,051276823	0,039761445
<b>Promedio Porcentaje SINAD</b>				
[%]	2,671112	4,403859	4,236429	4,218207
<b>Valor Máximo Porcentaje SINAD</b>				
[%]	2,727751	4,442198	4,258987	4,232193
<b>Valor Mínimo Porcentaje SINAD</b>				
[%]	2,636681	4,397865	4,227151	4,211396

Tabla 11: Medidas SINAD utilizando el algoritmo runVar para la Prueba Integral I

El Algoritmo runVAR fue el que presentó una menor variación en el valor de SINAD.



Cada uno de los valores de las medidas en las tablas comparativas fue tomado como el promedio de 150 muestras. Las tablas que contienen todos estos datos se encuentran en los Anexos Electrónicos<sup>30</sup>.

## 6.7 Prueba integrada 2

Esta prueba sigue la misma metodología que la prueba integrada 1, pero para este caso se va a medir el SINAD de una señal que se compone de la suma de otras señales; así que aparte de medir el ruido y la distorsión de la señal, también se va a medir el efecto de la intermodulación.

Las señales generadas por el programa Goldwave tenían las siguientes especificaciones:

*Señal saturada al 90%*

*Amplitud= 1 Vp*

*f<sub>i1</sub>=1009 Hz*

*f<sub>i2</sub>=1039 Hz*

*f<sub>s</sub>(programa)=44100 Hz*

Las señales generadas por Goldwave se observaron en el osciloscopio:

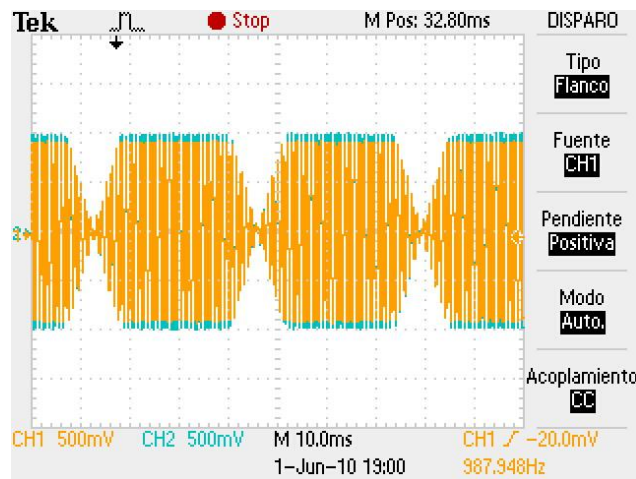


Figura 28: Señales de entrada prueba integrada 2

La señal en amarillo es el canal izquierdo (o la deseada)

La señal en azul es el canal derecho (o la entrada)

Luego de esto las señales fueron procesadas por el códec de la tarjeta con las siguientes especificaciones:

*f<sub>s</sub>(procesamiento en CCS)=32 kHz*

<sup>30</sup> El archivo que contiene estas pruebas es el siguiente: PRUEBA\_GOLD\_SAT.xls

Número de muestras tomadas=2757

Después de haber pasado la señal al *workspace* así se ve en MATLAB:

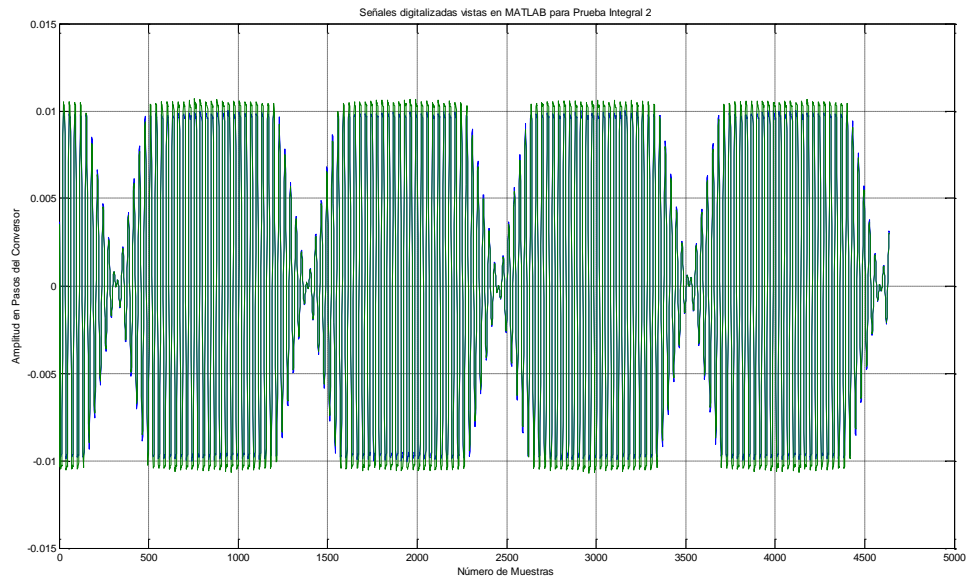


Figura 29: Señales digitalizadas vistas en MATLAB para Prueba Integral 2

Para trabajar las señales importada al *workspace* de MATLAB en Simulink se siguió el mismo procedimiento de la Prueba Integral 1.

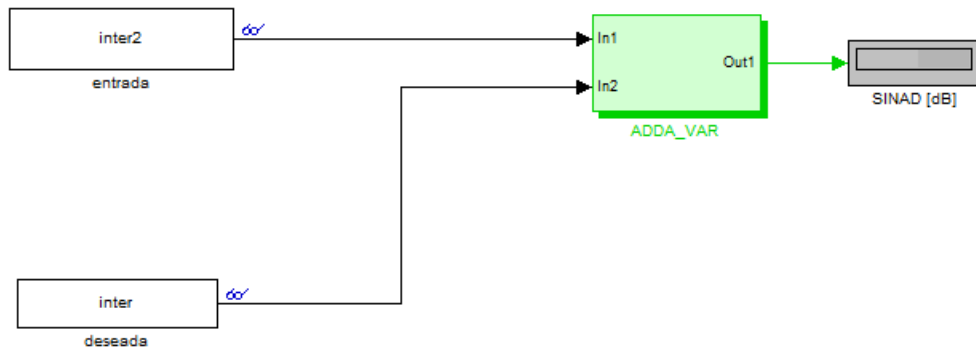


Figura 30: Simulación Prueba Integrada 2 utilizando señales externas

### Algoritmo VAR en Simulink usando

$$u=4$$

$$N=4$$

Entregó un valor de SINAD=27,01 dB

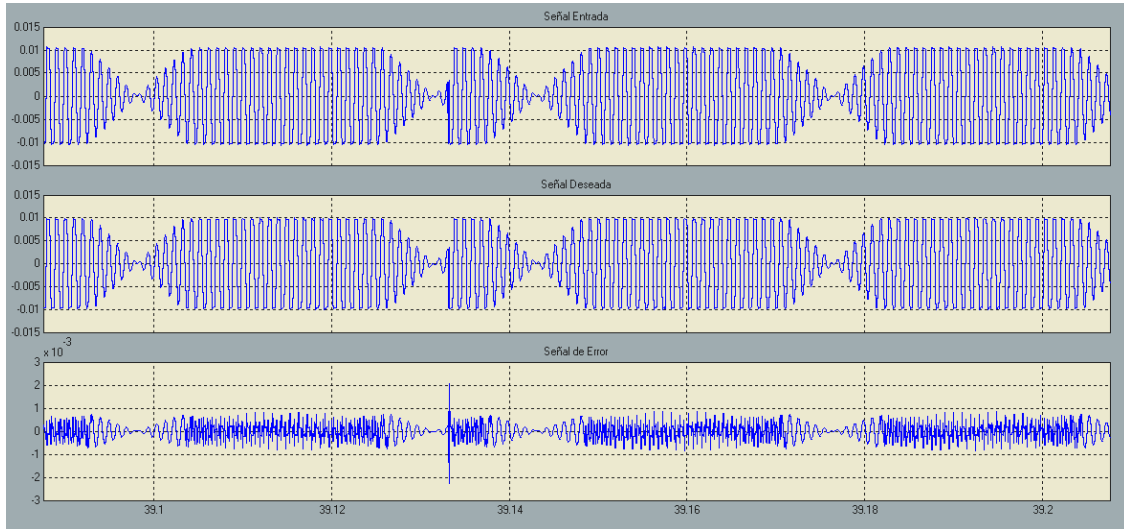


Figura 31: Zoom de Señales en tiempo (Entrada, Deseada y Error) para Prueba Integrada 2 utilizando algoritmo VAR

En las gráficas se puede observar un salto en los valores, esto se debe a que no se tomaron ciclos completos de las señales en la adquisición y al usar la propiedad cíclica este tipo de problemas se encuentran. Además, mirando la Señal de Error se puede observar la manera como se adapta el algoritmo; esta característica es bastante importante porque diferencia al instrumento ADDA de todos los demás Analizadores de Distorsión Armónica, debido a que con este instrumento se puede utilizar cualquier tipo de onda como señal de referencia. Asimismo, el instrumento ADDA es capaz de medir la intermodulación.

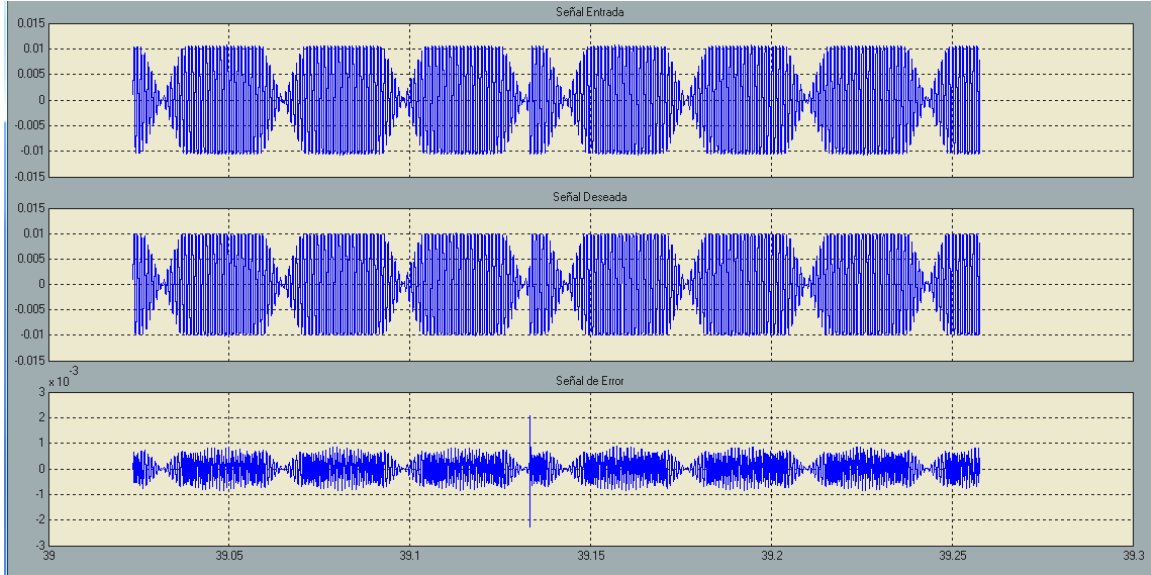


Figura 32: Señales en tiempo (Entrada, Deseada y Error) para Prueba Integrada 2 utilizando algoritmo VAR

El salto de valores es más apreciable en esta gráfica. Este tipo de saltos inducen un mayor ruido en las señales que se refleja en un valor de SINAD menor.

### Algoritmo runVAR en Simulink usando

$$u=4$$

$$N=4$$

$$\text{SINAD}=23,14 \text{ dB}$$

No se muestran las gráficas de las señales en tiempo del Algoritmo runVAR, porque son exactamente las mismas que las del Algoritmo VAR, debido a que la etapa de filtrado adaptable es la misma.

Con las mismas señales que se utilizaron para evaluar las simulaciones en Simulink, se hicieron varias medidas con el Instrumento ADDA. Las tablas que contienen todos los datos que sirvieron para promediar los valores de las siguientes dos tablas, se encuentran en los Anexos Electrónicos<sup>31</sup>.

	Medidas SINAD utilizando el algoritmo VAR		
	u=1E-10	u=1E-11	u=1E-12
<b>Promedio SINAD</b> [dB]	30,398069	28,887967	26,935688
<b>Valor Máximo SINAD</b> [dB]	30,976612	29,518290	27,812019
<b>Valor Mínimo SINAD</b> [dB]	29,988495	27,956472	24,870789
<b>Porcentaje Variación SINAD</b> [%]	0,660005442	0,96721404	2,121138701
<b>Promedio Porcentaje SINAD</b> [%]	3,021420	3,596050	4,510029
<b>Valor Máximo Porcentaje SINAD</b> [%]	3,166469	4,001072	5,707693
<b>Valor Mínimo Porcentaje SINAD</b> [%]	2,825982	3,342608	4,068169

Tabla 12: Medidas SINAD utilizando el algoritmo VAR para la Prueba integral 2

<sup>31</sup> El archivo que contiene estas pruebas es el siguiente: PRUEBA\_GOLD\_INTER.xls

Como se puede apreciar, las medidas de SINAD obtenidas con el instrumento ADDA son bastante mayores a las obtenidas en las simulaciones en Simulink. Esto se debe al comportamiento observado en las gráficas en tiempo; como se tomaron periodos incompletos de las señales, cuando se cumple la función cíclica, se van a encontrar picos que van a afectar la medida de SINAD disminuyéndola gracias al ruido que los picos inducen en la señal.

Medidas SINAD utilizando el algoritmo VAR			
	u=1E-10	u=1E-11	u=1E-12
<b>Promedio SINAD</b> [dB]	29,552512	28,350212	26,399845
<b>Valor Máximo SINAD</b> [dB]	29,774057	28,610392	26,889605
<b>Valor Mínimo SINAD</b> [dB]	29,415125	28,133934	24,208445
<b>Porcentaje Variación SINAD</b> [%]	0,204212973	0,502729581	1,445668729
<b>Promedio Porcentaje SINAD</b> [%]	3,329544	3,824259	4,792215
<b>Valor Máximo Porcentaje SINAD</b> [%]	3,382547	3,920156	6,159958
<b>Valor Mínimo Porcentaje SINAD</b> [%]	3,245616	3,710910	4,523971

Tabla 13: Medidas SINAD utilizando el algoritmo runVAR para la Prueba Integral 2

Para esta prueba, también se observó el valor óptimo  $u=1E-10$ .

Para validar las medidas del Instrumento ADDA se hicieron unas simulaciones extras con señales ideales creadas en Simulink:

***Intermodulación en Simulink caso ideal:***

*En esta simulación se utilizaron los siguientes valores:*

$$N=4$$

$$u=0.1$$

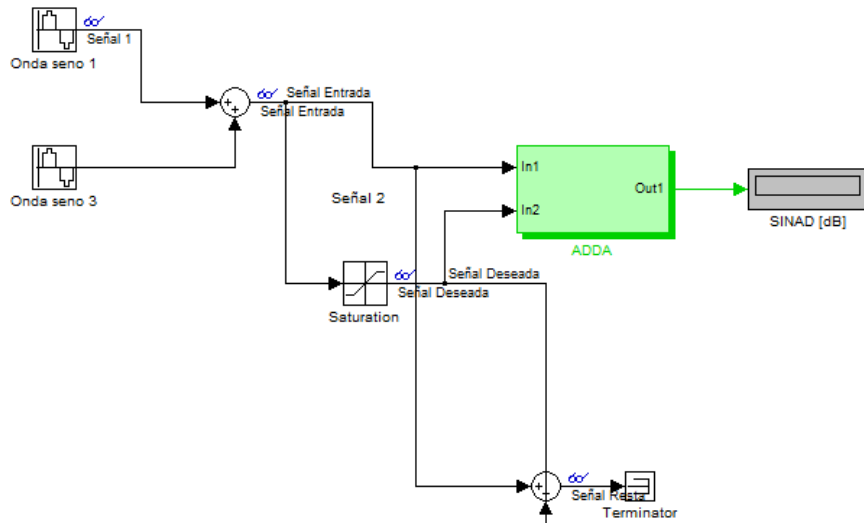


Figura 33: Simulación ADDA en MATLAB con una suma de señales como señal de entrada

Y el valor entregado por la simulación fue el siguiente:

SINAD=32,25 dB

Este valor corrobora los valores entregados por el instrumento ADDA, y confirma que de hecho los picos existentes en la simulación anterior, indujeron ruido a la medida.

Gráficos en tiempo de la simulación:

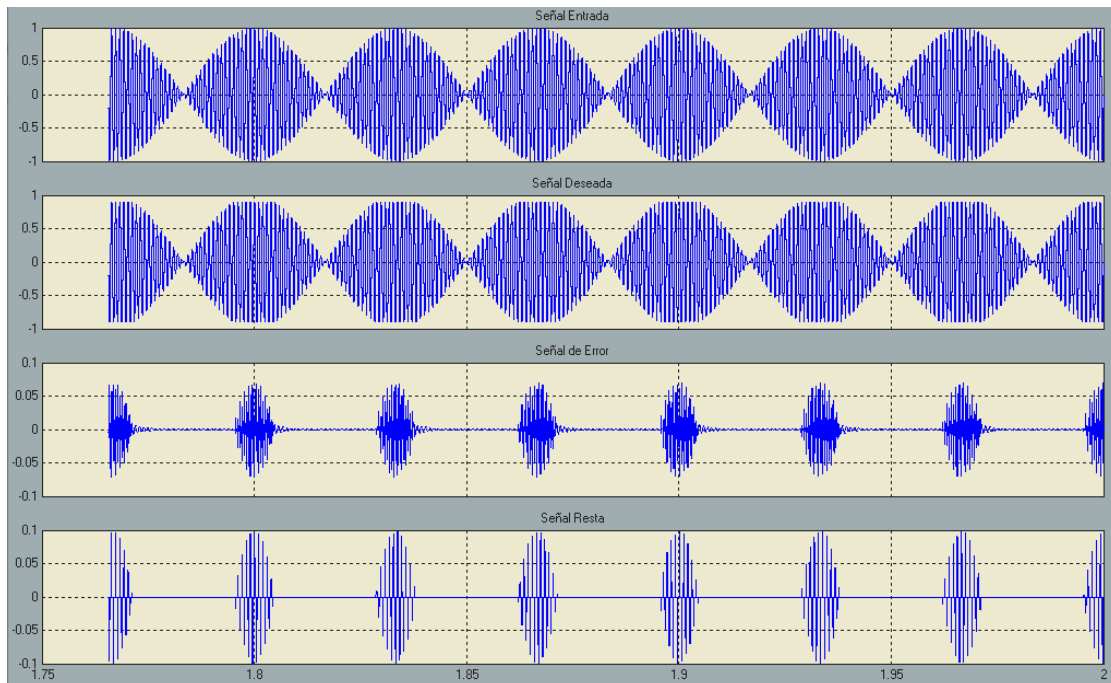


Figura 34: Señales en tiempo (Referencia, Suma, Error, Resta) Simulación ADDA con una suma de señales como señal de entrada

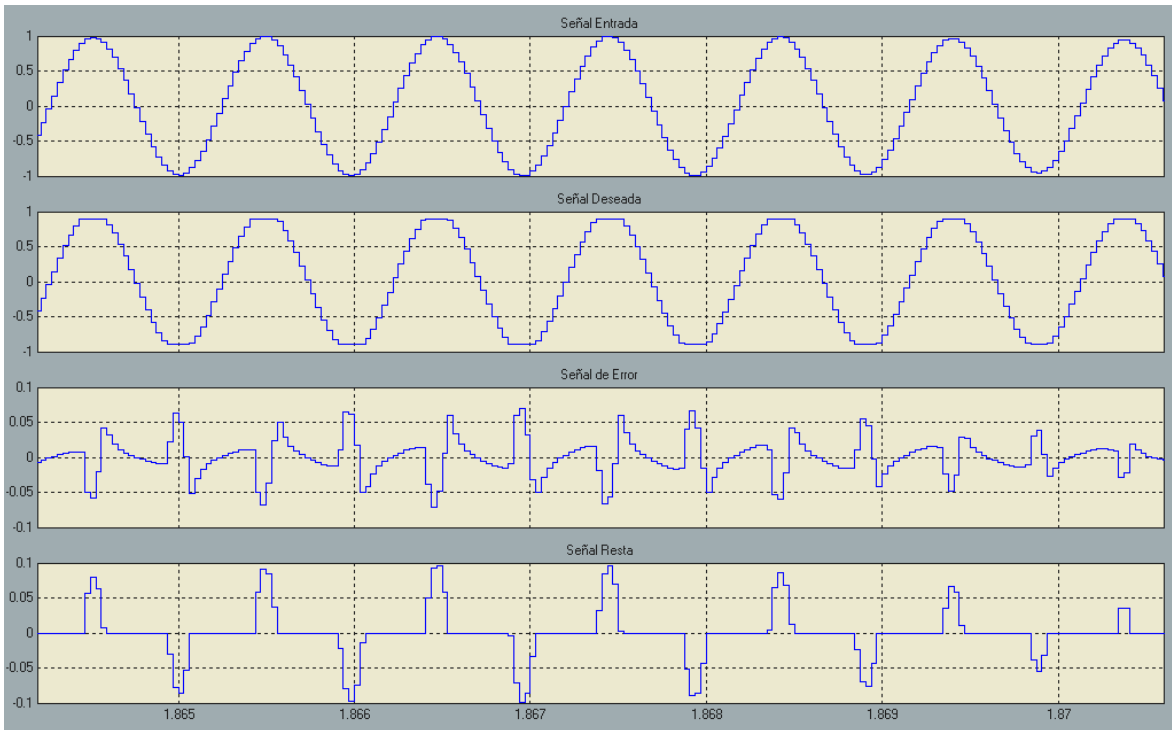


Figura 35: Zoom Señales en tiempo (Referencia, Suma, Error, Resta) Simulación ADDA con una suma de señales como señal de entrada

En la gráfica en la que se hace zoom se puede observar la diferencia entre la Señal de Error y la Señal de Resta entra las dos señales de entrada. Cuando la diferencia es negativa el algoritmo se adapta dando un pico positivo.

Estas Pruebas Integrales fueron motivadas en gran parte por la diferencia tan grande que existe entre los valores de  $u$  de las simulaciones de Simulink con respecto a los valores de  $u$  que son usados en el Instrumento ADDA. Debido a que el valor del parámetro  $u$  depende de la naturaleza de la señal de entrada (en particular se relaciona con los valores propios de matriz de correlación de la señal de entrada), se pensó que si se trabaja con las mismas señales a la entrada del filtrado adaptable, los filtros iban a trabajar con valores de  $u$  de órdenes de magnitud cercanos. Este no fue el caso, pero se puede deber a que la manera como se hacen los cálculos en Simulink es muy distinta a los métodos que se utilizan en el DSP, causando así que la naturaleza de la señal de entrada cambie. De todos modos, se verificó la validez de los algoritmos y la adquisición de datos, al observar las señales y al ver los valores de SINAD entregados por las simulaciones.

## 7 CONCLUSIONES

### **Comportamiento Frecuencia de Muestreo (fs)**

La disminución del valor de SINAD que sucede al aumentar la frecuencia de muestreo (fs) se debe al conversor Sigma-Delta del códec. Como el conversor Sigma-Delta promedia, a medida que aumenta la fs, menos muestras tiene para hacer el promedio así que esta estimación se genera con un mayor nivel de ruido, explicando así un valor de SINAD menor para la misma señal de entrada. De todos modos la variación no es muy significativa en el rango de señales de audio.

En la simulación teórica de MATLAB La variación del valor de SINAD con respecto a la variación de la frecuencia de muestreo (fs) es directamente proporcional; este comportamiento se da porque al aumentar la frecuencia de muestreo, se tiene una mayor resolución y como en esta simulación no existe conversión A/D, el valor de SINAD no depende del códec ni del tipo de conversión que ésta usa. Por esta razón, el aumento de la frecuencia de muestreo fs se da más como un beneficio en las simulaciones de MATLAB, mientras que para el Instrumento ADDA es una limitación en desempeño.

### **Comportamiento Frecuencia Fundamental Señal de Entrada (fi)**

La frecuencia de la señal de entrada (fi) presenta una estrecha relación con la frecuencia de muestreo (fs). Para que el Instrumento ADDA entregue medidas acertadas, la frecuencia de muestreo debe ser por lo menos 3,2 veces el valor de la frecuencia de la señal de entrada. Esta característica plantea la opción de trabajar en rangos de frecuencia y en ese caso se podría obviar la frecuencia de muestreo de 32k porque las demás cubren ese rango y dejar las de los extremos para tomar más valores; utilizando así las frecuencias de muestreo fs=8kHz, fs=16kHz, fs=48kHz, fs=96kHz.

Para un uso óptimo del instrumento ADDA, el valor de la frecuencia de muestreo fs podría colocarse como un parámetro de variación por parte del usuario, dependiendo de la frecuencia fundamental de la señal de entrada o del valor de la frecuencia mayor de la frecuencia de entrada, para el caso en el que la señal de entrada sea una suma de señales con diferentes frecuencias.

### **Algoritmo VAR versus Algoritmo runVAR**

Los dos algoritmos VAR y runVAR entregan valores estables y se comportan de una manera similar. De todos modos, en términos de recursos tienen diferencias importantes.

El Algoritmo VAR necesita almacenar la cantidad de muestras del registro para dos señales, y además debe recorrer los registros dos veces, por cada señal, para entregar una medida de SINAD. En cambio el Algoritmo runVAR sólo necesita el número de muestras del registro para la primera medida y de esa medida en adelante, para cada muestra entrega un valor de SINAD; de este modo, el analógico del registro del Algoritmo VAR sería una ventana para el Algoritmo runVAR. Debido a lo anterior, si se decidiera tener un registro de un valor mucho mayor a 2048, se podrían ver diferencias significativas de desempeño entre el Algoritmo VAR y el Algoritmo runVAR.

Asimismo, el algoritmo runVAR se podría optimizar aplicando factores de peso que decrecen de modo exponencial. Esto haría que las muestras más antiguas perdieran peso de modo exponencial, dando así más importancia a muestras más recientes.



Actualmente el Algoritmo runVAR diseñado con una cota de 20048 muestras. Esta cota superior impide que el Algoritmo runVAR trabaje completamente en tiempo real como se podría tomar. De todos modos, a diferencia del Algoritmo VAR este no debe almacenar registros, así que una implementación mucho más óptima del Algoritmo VAR, sería el Algoritmo runVar como se encuentra ahora mismo. Si el Algoritmo runVAR se dejara sin cota superior, funcionaría perfectamente porque no se presentaría *overflow*<sup>32</sup> de los valores de la varianza; la única limitante que tendría sería el valor máximo que pueda alcanzar la variable que indica el número de interrupciones.

### **Limitación Resolución del Códec**

El valor máximo que puede entregar el instrumento ADDA se ve notoriamente limitado a la resolución del Códec. Sería conveniente implementar el instrumento ADDA en una tarjeta con un Códec de por lo menos 20 ENOB para tener una resolución mayor a la de los Analizadores de Distorsión existentes en el laboratorio.

### **Optimizaciones Propuestas para el Instrumento**

El instrumento ADDA puede tener varias optimizaciones en su diseño. Las siguientes son las propuestas:

Implementar una estimación del número de pesos del filtro (N) en tiempo real. Esto se podría lograr analizando la sensibilidad de la Señal de Error con respecto a N.

Estimar el valor óptimo del parámetro de tamaño de paso  $u$ , por medio de la matriz de correlación de la señal de prueba y sus valores propios.

Evitar la necesidad de una señal de referencia estimando la frecuencia fundamental de la señal de prueba [23] , si es una señal armónica.

### **Medida de Señales Arbitrarias**

El instrumento ADDA está en capacidad de medir la distorsión y el ruido de cualquier tipo de onda, haciendo uso de una señal de referencia que no es precisamente una señal con una frecuencia fundamental a diferencia de sus instrumentos equivalentes analógicos. Esto se vio demostrado en la Prueba Integrada 2. Adicionalmente, el instrumento ADDA puede medir la cantidad de intermodulación.

---

<sup>32</sup> desbordamiento

## 8 BIBLIOGRAFÍA Y FUENTES DE INFORMACIÓN

- [1] *Leader LDM-171 : Semi-Automatic Distortion Meter*, Hoja de Especificaciones.
- [2] Cooper, William D. Helfrick Albert D. *Instrumentación Electrónica Moderna y Técnicas de Medición*. Ed. Prentice Hall Hispanoamericana, S.A., 1991. ISBN 968-880-236-0
- [3] <http://www.klippel.de/analyzer/default.asp>
- [4] <http://www.sencore.com/markets/a-v-installation-calibration/14>
- [5] <http://sewelldirect.com/Sencore-DigiPro-Digital-Audio-Analyzer.asp>
- [6] Turley, Andrew C. Stikvoort, Eduard. *Real-Time Digital Distortion Analyzer*. Philips Research Labs., Eindhoven, The Netherlands
- [7] *Hameg HM 8027: Distortion Meter*, Hoja de Especificaciones.
- [8] <http://www.goldwave.com/>
- [9] Pohlmann, Ken C. *Principles of Digital Audio* 4<sup>th</sup> Edition. Ed. McGraw-Hil, 2000. ISBN 0-07-134819-0
- [10] Chassaing, Rulph. Reay, Donald. *Digital Signal Processing and Applications with the TMS320C6713 and TMS320C6416 DSK*. Second Edition. Ed. Wiley-Interscience. ISBN 978-0-470-13866-3
- [11] Widrow, Bernand. Stearns, Samuel D., *Adaptive Signal Processing*, Ed. Prentice-Hall PTR, 1985. ISBN 0-13-004029-0
- [12] Haykin, Simon, *Adative Filter Theory* 3<sup>rd</sup> Edition,Ed. Prentice Hall,1996. ISBN 0-13-322760-X
- [13] Kester, Walt. *Understand SINAD, ENOB, SNR, THD, THD + N, and SFDR so You Don't Get Lost in the Noise Floor*. Analog Devices. MT-033 TUTORIAL.
- [14] Knuth, Donald E. *Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Third Edition.
- [15] Chan, Tony F.; Golub, Gene H.; LeVeque, Randall J. (1983). *Algorithms for Computing the Sample Variance: Analysis and Recommendations*. The American Statistician 37, 242-247.
- [16] Ling, Robert F. (1974). *Comparison of Several Algorithms for Computing Sample Means and Variances*. Journal of the American Statistical Association, Vol. 69, No. 348, 859-866.
- [17] *Stratix II DSP Development Board Reference Manual*. ALTERA. Document Version 6.0.1.
- [18] Petrone, Joseph. *Adaptive Filter Architectures for FPGA Implementation*.
- [19] <http://focus.ti.com/docs/toolsw/folders/print/tmdsdsk6713.html>
- [20] <http://www.altera.com/products/devkits/altera/kit-dsp-2S60.html>
- [21] Texas Instruments, *TMS320C6416/C6713DSK One-Day Workshop, Student Guide*

- [22] Texas Instruments, *TMS320C600 Code Composer Studio Tutorial*.
- [23] S. Kim and y. Park, *On-Line Fundamental Frequency Tracking Method for Harmonic Signal and Application to ANC*.
- [24] Texas Instruments, *TLV320AIC23 Stereo Audio Codec, 8 to 96kHz, With Integrated Headphone Amplifier*.