

## **SIDESIM-O: HERRAMIENTA DE SIMULACIÓN DE MODULADORES $\Sigma\Delta$ BASADA EN “OCTAVE”**

***Ximena Victoria Bouzas Munguía***

Universidad Autónoma Metropolitana, Unidad Azcapotzalco  
*ximebomu@hotmail.com*

***Javier Alducin Castillo***

Universidad Autónoma Metropolitana, Unidad Azcapotzalco  
*ximebomu@hotmail.com*

***Juan Jesús Ocampo Hidalgo***

Universidad Autónoma Metropolitana, Unidad Azcapotzalco  
*ximebomu@hotmail.com*

***Iván Vázquez Álvarez***

Universidad Autónoma Metropolitana, Unidad Azcapotzalco  
*ximebomu@hotmail.com*

### **Resumen**

En este trabajo se presenta el desarrollo de una herramienta para la simulación a nivel sistema de moduladores Sigma-Delta de alto orden y de un solo bit realizado en tiempo discreto usando técnicas de capacitor conmutado. En contraste con otros métodos, aquí se propone el uso de software libre, en particular del programa “Octave”, como plataforma para crear el simulador. El funcionamiento del programa desarrollado se compara, mediante la simulación de dos arquitecturas, con el de un programa comercial, obteniéndose resultados satisfactorios. El uso de software libre evita problemas tales como la piratería de programas o la adquisición de licencias de alto costo.

**Palabra(s) Clave(s):** Conversión A/D, Modulación Sigma-Delta, Simulación a nivel sistema

## 1. Introducción

Desde hace muchos años existe la tendencia en utilizar señales digitales en lugar de analógicas, ejemplo de ello fue diciembre de 2015, mes en que México sufrió el llamado “apagón analógico”, a partir de entonces, las televisiones que captan señales analógicas han dejado de ser capaces de desmodular las señales de televisión, ya que ahora son digitales. Para que estas televisiones puedan decodificar dichas transmisiones es necesario agregar unos dispositivos que conviertan la señal y así volver a hacer útil al televisor.

Las señales de información provenientes de sensores o transductores de audio, nivel de luz, PH, presión, temperatura, etc., son del tipo analógico, las cuales, se deben de modificar para convertirlas en una señal digital y de esta manera poder ser procesadas digitalmente. El tratamiento digital de las señales permite expresar una propiedad o un atributo físico de la misma, facilitando el cálculo numérico de diversos parámetros existentes en cada conjunto de señales, a través del uso de procesadores digitales. Para realizar la conversión de una señal analógica a digital es necesaria la implementación de un convertidor analógico-digital (ADC).

Existen principalmente dos familias de convertidores, la primera de ellas es la de ADC's que funcionan a la frecuencia de muestreo indicada por el teorema de Nyquist. El otro tipo de ADC es aquel que realiza un muestreo muy superior al indicado por la frecuencia de Nyquist, a este tipo de convertidores se les llama ADC Sigma-Delta ( $\Sigma\Delta$ ).

Los ADC  $\Sigma\Delta$  están basados en tres operaciones:

- Sobre muestreo: En estos ADC's se utiliza una frecuencia de muestreo que es usualmente 128, 256, o 512 veces mayor que la frecuencia de Nyquist. A este factor de aumento en la frecuencia de muestreo se le llama *factor de sobre muestreo* (OSR).
- Cuantificación de la señal: La cuantificación es una operación inherente a cualquier ADC. Es un proceso no-lineal y no-reversible, donde un intervalo continuo de valores de amplitud, es transformado en un conjunto finito de códigos digitales de salida. Aquí se introduce una pérdida de información que se modela como un ruido de cuantificación.

- Conformado de ruido: es un filtrado del ruido de cuantificación en la banda de interés.

Debido al uso del sobre muestreo y conformado de ruido, se tiene una mejora significativa de la razón señal a ruido (SNR) del sistema dentro del ancho de banda útil del convertidor, por esta razón, este tipo de convertidores son una alternativa efectiva para convertir señales con un alto número de bits. Para un ADC  $\Sigma\Delta$  ideal, de orden  $n$ , la SNR está dada por la ecuación 1.

$$SNR = \frac{3(2n+1)OSR^{2n+1}}{2\pi^{2n}} \quad (1)$$

Conforme fue aumentando el uso de los ADC  $\Sigma\Delta$ , también fue creciendo el uso de herramientas computacionales para la simulación, diseño y análisis de estos.

Dada la naturaleza de los moduladores  $\Sigma\Delta$ , cuya base es el sobre muestreo y el conformado de ruido, si dicha simulación se llevase a cabo en un programa de simulación de circuitos tal como lo es Spice (ver figura 1), se requeriría de un gran tiempo de análisis por computadora: Spice ejecuta su simulación a nivel transistor, es decir, para poder generar la ecuación de algún bloque de circuito se usan todos los elementos que lo conforman. Por ejemplo, un interruptor CMOS, el cual utiliza cuatro transistores, luego entonces, el programa produciría cuatro ecuaciones para modelarlo, en lugar de una. En la figura 1 se observa el diagrama esquemático conceptual de un modulador  $\Sigma\Delta$  de primer orden, realizado usando técnicas de capacitor conmutado (SC). Como se ve, el circuito consta de cuatro interruptores, un amplificador operacional (OPAMP) y un comparador (COMP).

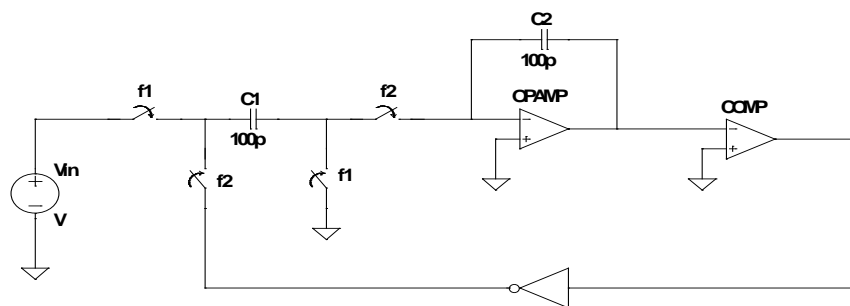


Figura 1 Diagrama esquemático de un modulador  $\Sigma\Delta$  de primer orden en SC.

Como se puede observar en la figura 2, un circuito comparador típico (LM111) consta de 24 transistores, más un OPAMP que tiene alrededor de 20 transistores y 16 transistores de los cuatro interruptores, esto se traduce en 60 ecuaciones, que se tendrán que analizar en cada incremento de tiempo del análisis transitorio que se realiza.

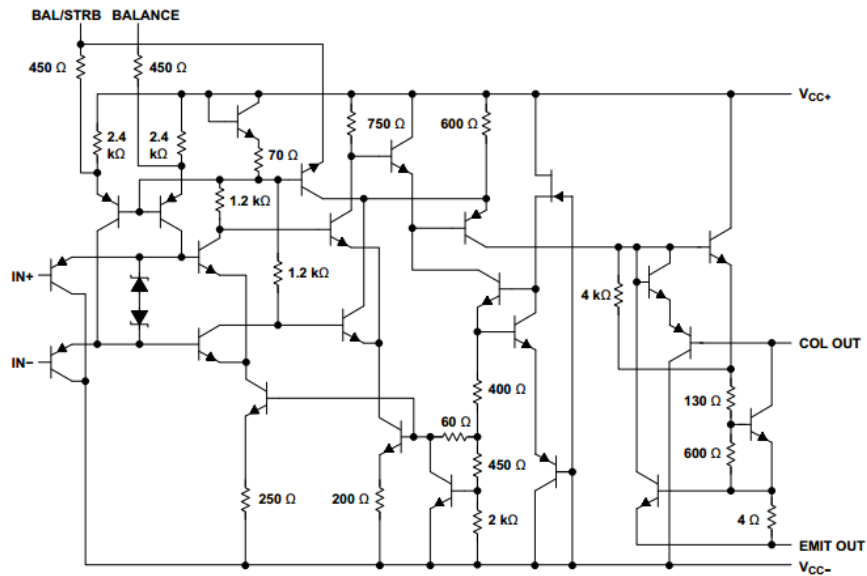


Figura 2 Diagrama esquemático interno del comparador LM111 [1].

Para realizar el análisis de la respuesta en frecuencia del modulador, se calcula la Transformada Discreta de Fourier (DTF) de la señal de salida, esto requiere un tiempo mínimo de simulación equivalente al de por lo menos 32 ciclos de la señal de entrada. Retomando el hecho del gran número de ecuaciones que se tiene que analizar, aunado al número de ciclos de entrada requerido por la DTF, tenemos como consecuencia un tiempo de simulación que puede llegar a ser de días, además de que los archivos de salida ocuparían una gran cantidad de memoria. Por las razones anteriormente expuestas se propone el uso de simulaciones de alto nivel, para realizar la adecuada modelación de este tipo de sistemas, las cuales, además de acelerar considerablemente el proceso de simulación, permiten analizar por separado los efectos que introducen las características no ideales de los elementos que conforman al modulador.

El desarrollo de herramientas de software destinadas a simular o analizar las señales obtenidas a través de los convertidores  $\Sigma\Delta$  está dominado por programas comerciales tal como la plataforma *MATLAB* de Mathworks.

En la literatura se encuentran varios trabajos relacionados: En [2] y [3] se propone una interfaz gráfica para el modelado de un ADC  $\Sigma\Delta$ , haciendo uso del toolbox de *MATLAB* y *SIMULINK*, el primero contiene una sola arquitectura de segundo orden, dicha interfaz es capaz de realizar la simulación de la arquitectura obteniendo el espectro de densidad de potencia (PSD), la estimación de SNR y ENOB, así como el tiempo transcurrido de análisis. Mientras que el segundo la interfaz calcula y corrige los valores para el vector del tiempo discreto, además de valores no ideales del tiempo continuo, el comportamiento real del OPAMP, jitter, exceso de retardos, etc.

En [4], se hace uso de *SIMULINK* para el modelado de un modulador de tercer orden, en aplicaciones de audio. El modulador analizado es de tipo no lineal, ya que adicionan ruidos debidos a efectos como slew rate, jitter, thermal noise, ganancia en DC y ruido blanco. En [5], se realiza una serie de bloques para modelar un ADC  $\Sigma\Delta$  de un circuito CMOS de  $0.5\mu\text{m}$ , se realizan dos simulaciones: una con condiciones ideales y otra con ruidos de comportamiento no lineal.

Se conoce que en la implementación de un modulador en capacitor-conmutado (SC), se usan diversas configuraciones de Amplificadores Operacionales (OPAMPS), los cuales tienen características tales como: ruido referido a la entrada, ganancia finita, ancho de banda finito, máxima razón de cambio y saturación de voltaje a la salida. Todo esto causa que el modulador  $\Sigma\Delta$  funcione de forma no ideal. Con base en lo anterior, en [6] se desarrolló un modelo en *SIMULINK* capaz de analizar las imperfecciones introducidas por los OPAMPS, lo cual se modeló en un modulador  $\Sigma\Delta$  de segundo orden. Se presentan como elementos de salida, gráficas de la SNR en función de cada uno de los parámetros de los OPAMPS.

En [7] se propone una serie de bloques creados en *SIMULINK* de un ADC  $\Sigma\Delta$  de segundo orden; para adicionar ruidos de efectos no lineales y obtener un modelado completo. Además, en [8] se modela un modulador de tercer orden

utilizando MATLAB, donde se compara cada integrador y se observa que no hay variación relevante. En [9], se hace uso de SIMULINK para el estudio de este trabajo, modelando el ruido que presenta el OPAMP cuando se tiene un modulador  $\Sigma\Delta$  utilizando el método en SC.

Otro trabajo relacionado es [10], donde se realiza el estudio de una arquitectura de alto orden para sensores de medición, utilizando la técnica de cuantificación de dos pasos. Los bloques que se modelaron contienen ruidos como lo son el ruido blanco, ganancia de corriente, tensiones de saturación, etc; así como diversas no idealidades, todo el proceso de cuantificación fue diseñado y simulado en MATLAB.

Como se observa, el modelado de alto nivel de este tipo de sistemas es común, pero hasta el momento, no se ha reportado la simulación del modulador  $\Sigma\Delta$  en un ambiente de software libre, ya que como se puede observar todos los métodos de modelado usan MATLAB, este tipo de plataformas brindan al usuario una gran cantidad de herramientas para el desarrollo de aplicaciones para análisis y simulación de sistemas de procesamiento de señales, sin embargo, adquirir una licencia con todos los módulos necesarios para el adecuado análisis de estos sistemas (en ambas plataformas) representa un alto costo.

*Octave* es un programa que contiene un lenguaje interpretado de alto nivel, destinado esencialmente a los cálculos numéricos. Dicha herramienta se encuentra dentro de la lista de paquetes de GNU. Este software brinda la capacidad de visualizar gráficas y manipular datos. Es utilizado normalmente a través de su interfaz de líneas de comandos, ya que no tiene un entorno visual donde se pueda programar, pero es esta misma falta de ventana de trabajo, la que permite que el procesamiento de datos sea muy rápido.

En este trabajo se presenta el desarrollo de un conjunto de primitivas que permiten el modelado de un modulador  $\Sigma\Delta$  dentro del ambiente de trabajo *Octave*, su principal aportación radica en brindar una alternativa sin costo para la simulación de estos moduladores, además es expandible, es decir, se pueden agregar funciones para realizar algún análisis específico requerido. Otro aspecto importante es que el tiempo de respuesta es reducido (a diferencia del modelado

en programas como Spice), con lo cual, se puede simular en un tiempo razonablemente pequeño, una cantidad de ciclos de entrada tal que, la estimación de la DFT es confiable, y consecuentemente se pueden obtener buenos parámetros de SNR, la razón señal a ruido más distorsión (SNDR), y el número efectivo de bits (ENOB) a partir del análisis espectral. Como objetivo inicial, se presenta una herramienta de simulación de moduladores  $\Sigma\Delta$  ideales (*SIDESIM-O*), realizados en SC, de alto orden y un solo bit.

## 2. Desarrollo

En la figura 3 se muestra un diagrama a bloques con las etapas del desarrollo de la herramienta computacional. El principal objetivo es fungir como ayuda en el análisis y diseño de moduladores  $\Sigma\Delta$ , a partir de la lectura de un archivo creado por el usuario que describe la arquitectura del modulador que sea desea analizar, obteniendo los datos del comportamiento en tiempo y frecuencia del mismo, posteriormente, se estiman las métricas de rendimiento SNDR, SNR y ENOB.

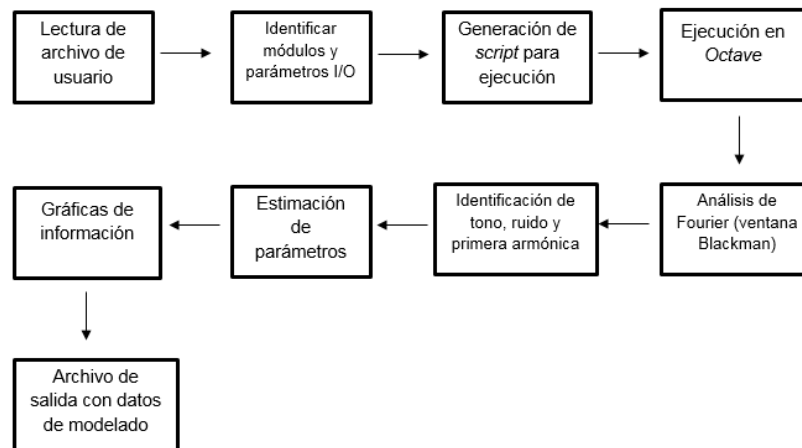


Figura 3 Diagrama de flujo de la metodología propuesta del simulador desarrollado.

Esta herramienta computacional fue desarrollada para modelar una arquitectura, donde el usuario debe escribir un archivo de texto, indicando los módulos a usar con los parámetros de entrada y de salida correspondientes. Los módulos disponibles son:

- Tono o señal de entrada (tono): en este bloque se genera una señal sinusoidal a una frecuencia y amplitud variable, que servirá para analizar la arquitectura que el usuario diseñe (figura 4).



Figura 4 Entradas y salida del módulo tono o señal de entrada.

- Restador (dif): bloque que realiza la diferencia de dos señales (figura 5), se utiliza principalmente en la diferencia de una señal retroalimentada.

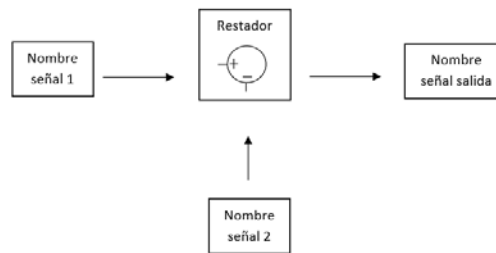


Figura 5 Entradas y salida del módulo restador.

- Constante (cte): realiza la ponderación o multiplicación por una constante (figura 6).

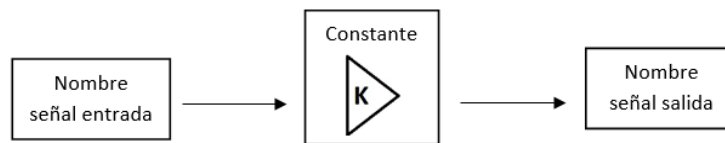


Figura 6 Entrada y salida del módulo constante.

- Integrador (int): realiza la integración en tiempo discreto de la señal (figura 7).

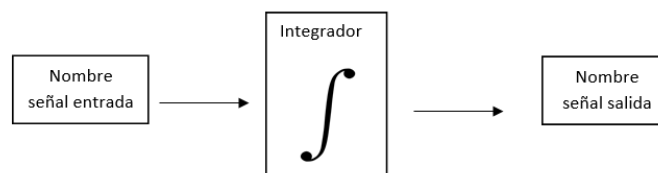


Figura 7 Entrada y salida del módulo integrador.



- Comparador (comp): en este bloque, se compara un valor de entrada ( $V_{EN}$ ) contra un valor de umbral ( $V_{UM}$ ) y determinar el valor de salida ( $V_{SAL}$ ) (figura 8) de acuerdo a la ecuación 2.

$$V_{SAL} = \begin{cases} 0, & V_{EN} < V_{UM} \\ 1, & V_{EN} \geq V_{UM} \end{cases} \quad (2)$$

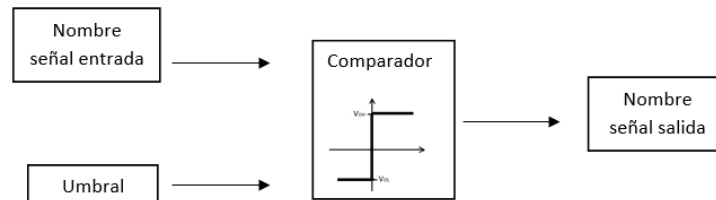


Figura 8 Entrada y salida del módulo comparador.

Produciendo valores de salida discretos que representan la salida del modulador  $\Sigma\Delta$ .

Otra operación necesaria en el diseño de moduladores de alto orden es la adición de señales, por lo tanto, se propone también crear un bloque sumador.

- Sumador (sum): realiza la suma de dos señales (ver figura 9), este bloque permite junto con el bloque de retardo crear esencialmente al integrador.

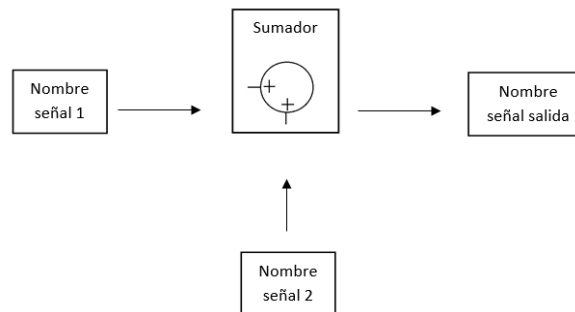


Figura 9 Entradas y salida del módulo Sumador.

El usuario, como en cualquier otro archivo de código, podrá ingresar todos los comentarios que requiera haciendo uso del símbolo '%', es necesario que el usuario guarde el archivo de descripción del modulador con una extensión “.ssd”, para que el texto se lea correctamente. Dicho archivo se lee, y a través de un intérprete, se identifican las funciones y los parámetros de entrada y salida de la arquitectura descrita por el usuario. El intérprete además reconoce y declara las

variables adicionales, necesarias para la correcta simulación del sistema, de tal forma que la identificación de módulos y su simulación se realizan sin la intervención del usuario.

Una vez que se identifican cada una de las funciones requeridas, la herramienta escribe un *script* que genera el código necesario en *Octave* para el modelado de la arquitectura, generando los resultados para su posterior análisis.

Para conocer el comportamiento en la frecuencia del modulador  $\Sigma\Delta$  propuesto, se analiza la salida a través de la transformada de Fourier usando el algoritmo Transformada Rápida de Fourier (FFT) utilizando la ventana de Blackman, ya que para los moduladores  $\Sigma\Delta$  ésta muestra resultados óptimos.

Posteriormente se identifica, dentro de la respuesta en frecuencia, la señal de tono ( $S_s$ ), la primera armónica ( $S_A$ ), y el ruido ( $S_N$ ) en banda, estos datos son necesarias para calcular los parámetros de rendimiento del modulador: SNR, SNDR y ENOB de acuerdo con las ecuaciones 3, 4 y 5 respectivamente.

$$SNR = 10 \log_{10} \left( \frac{\frac{1}{N} \sum_{n=0}^{N-1} S_S^2}{\sum_{n=0}^{N-1} S_{SN}^2} \right) \quad (3)$$

$$SNDR = 10 \log_{10} \left( \frac{\sum_{n=0}^{N-1} S_S^2}{\sum_{n=0}^{N-1} S_{SN}^2 + \sum_{n=0}^{N-1} S_{SA}^2} \right) \quad (4)$$

$$ENOB = \frac{SNDR - 1.76}{6.02} \quad (5)$$

Finalmente, la herramienta brinda un archivo de datos, el usuario puede elegir la información que desea conocer, por ejemplo: el tiempo de simulación, los datos de la salida de cada elemento, etc. Además de presentar el comportamiento en frecuencia de la arquitectura analizada, esto para que el usuario observe dicha respuesta y tenga además de las métricas de rendimiento, elementos para visualizar el comportamiento de la arquitectura.

### 3. Resultados

Para observar y mostrar que la herramienta *SIDESIM-O* cumple los objetivos planteados, se comparan los datos obtenidos a través de la herramienta

desarrollada con los datos obtenidos en SIMULINK (MATLAB) al simular dos arquitecturas en ambas plataformas.

La primera arquitectura es de primer orden (figura 10), este modulador fue simulado usando un tono con amplitud de 0.7245 volts, frecuencia de 61 Hz, frecuencia de muestreo ( $f_s$ ) de 250kHz, número de muestras ( $m$ ) de 1250000 (se simulan 30 ciclos del tono) y un nivel de offset de 0.75 volts. Suponiendo un ancho de banda de 125Hz, la  $f_s$  produce un  $OSR$  de 1000. Los elementos que conforman esta arquitectura son: un punto de resta, de la señal de entrada  $x(z)$  con la señal de salida  $y(z)$ , un integrador ideal y un comparador, donde el nivel bajo es 0 y el nivel alto es 1.5 teniendo como nivel de referencia 0.75 volts.

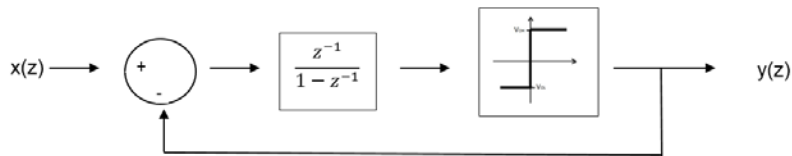


Figura 10 Diagrama a bloques de arquitectura de primer orden.

El procesamiento de datos, así como el análisis en frecuencia del modulador, fue realizado en 200 segundos, en un equipo de cómputo con un procesador Intel Quad Core de 2.4 GHz y 2 G de RAM. En la figura 11, se observa el comportamiento en frecuencia de la señal de salida de la arquitectura modelada, la gráfica de color azul representa la respuesta en frecuencia del modulador  $\Sigma\Delta$  modelado en *SIDESIM-O*, y dentro de la misma, la gráfica de color rojo es el mismo modulador simulado en SIMULINK. El comportamiento es idéntico, el ruido de baja frecuencia muestra niveles semejantes, la presencia del tono a 61 Hz, así como de armónicas y ruido en frecuencias altas coinciden plenamente.

En la tabla 1, se muestran las métricas de desempeño estimadas, se observa que la variación en SNR y SNDR, obtenido con SIMULINK y *SIDESIM-O* es de 0.108. El valor teórico de SNR se obtiene a partir de la ecuación 1, con  $n=1$  y  $OSR=1000$ . La estimación de las métricas se realizó, con el mismo script y los mismos parámetros, generando valores muy cercanos entre sí.

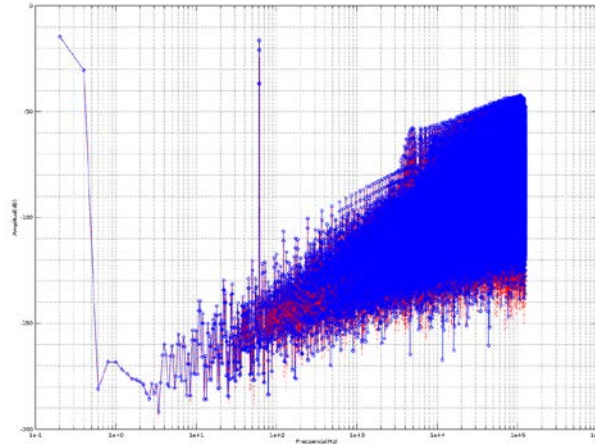


Figura 11 Frecuencia de la señal de salida de la arquitectura modelada.

Tabla 1 Parámetros de rendimiento de la arquitectura de primer orden.

Herramienta	SNR	SNR (teórico)	SNDR	ENOB
SIMULINK/MATLAB	89.166	88.808	89.166	14.519
Octave	89.224	88.808	89.224	14.529

Una segunda arquitectura que se modeló en ambos ambientes para su comparación es la que se observa en la figura 12.

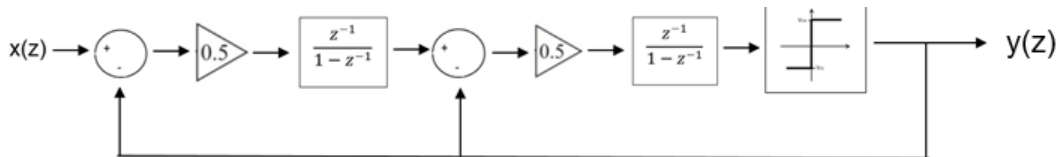


Figura 12 Diagrama a bloques de arquitectura de segundo orden.

Este modulador se simuló usando un tono con amplitud de 0.8 volts, frecuencia de 61Hz,  $f_s = 250\text{kHz}$ ,  $m=1250000$  (se simulan 30 ciclos) y un offset de cero volts, a diferencia del anterior, este es un modulador de segundo orden, ya que contiene dos integradores con retardo. Para el cálculo teórico de la SNR se asumió el mismo ancho de banda de 125Hz,  $n=2$  y OSR de 1000.

Al igual que en la primera arquitectura, en la figura 13, se muestra el comportamiento en frecuencia del modulador  $\Sigma\Delta$  de segundo orden, como era de esperarse la atenuación del ruido de cuantificación es mayor.

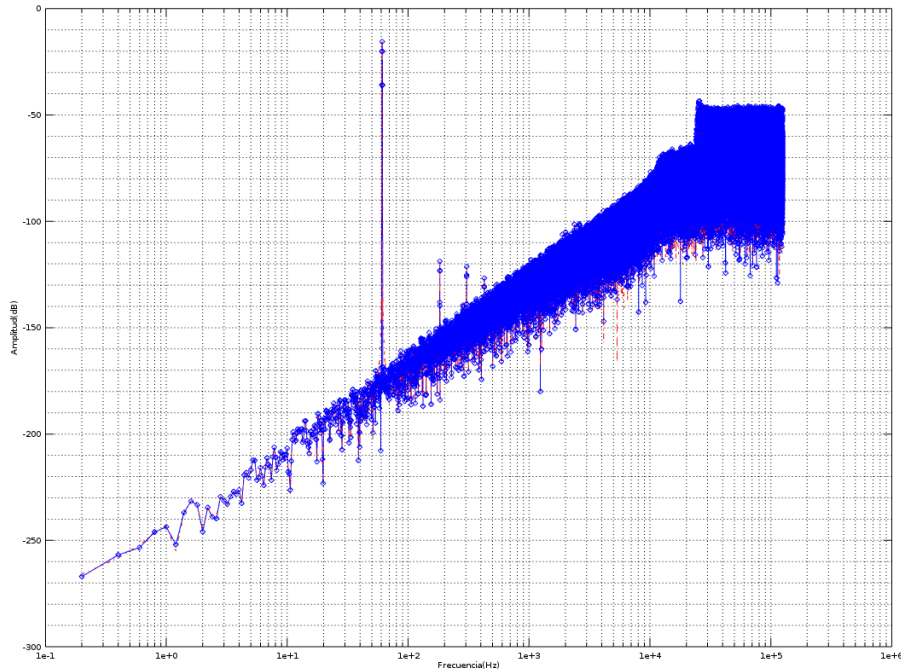


Figura 13 Respuesta en frecuencia de la arquitectura de segundo orden.

En la tabla 2, se comparan los parámetros de rendimiento para la arquitectura de segundo orden, se observa que las métricas obtenidas con la presente herramienta son superiores a las obtenidas con los datos obtenidos en SIMULINK, además el resultado del método propuesto, se acerca más el valor de SNR predicho por la ecuación 1, a su vez, el valor de ENOB, al depender del valor de SNDR, es mayor que el obtenido con SIMULINK.

Tabla 2 Parámetros de rendimiento arquitectura de segundo orden.

Herramienta	SNR	SNR (teórico)	SNDR	ENOB
SIMULINK/Matlab	120.73	138.86	120.73	19.763
Octave	128.74	138.86	128.74	21.094

#### 4. Discusión

Los datos obtenidos con la herramienta computacional *SIDESIM-O* desarrollada en este trabajo, son muy cercanos a los obtenidos en SIMULINK, tanto en frecuencia como en métricas de rendimiento para la arquitectura de primer orden. En el caso de la arquitectura de segundo orden, el comportamiento en frecuencia

es prácticamente idéntico (Fig. 13), sin embargo, en la estimación de las métricas de desempeño hay un incremento en el valor de los datos obtenidos con la herramienta de software libre en donde la SNR, SDNR y ENOB son superiores, con respecto a los datos estimados con SIMULINK. Esto puede ser indicio de que, en arquitecturas de alto orden, la simulación realizada en SIMULINK sea afectada por el redondeo de datos interno que maneja MATLAB, es posible solucionar el problema anterior al modificar las opciones de la configuración de redondeo y tolerancia absoluta en la simulación de SIMULINK en cada simulación.

Además, el valor de SNR calculado teóricamente con la ecuación 1, es más cercano con el estimado a partir de los datos de la herramienta propuesta, eso puede ser un indicio adicional de que los datos obtenidos con SIMULINK, sean afectados de manera interna por los ajustes de la máquina de simulación.

Es importante recalcar que, de acuerdo con los objetivos iniciales del trabajo, las arquitecturas simuladas son ideales, es decir, que no se ha incorporado ningún tipo de ruido de circuito, incertidumbre en el muestreo ni ganancia finita del amplificador operacional, es por eso que los valores de SNR y SNDR tanto en *SIDESIM-O* como en *SIMULINK* son relativamente altos, a diferencia de aquellos donde los ruidos son considerados en la simulación y cálculo de las métricas de desempeño.

## 5. Conclusiones

En este trabajo se presentó una herramienta de simulación de moduladores  $\Sigma\Delta$  de alto orden y de un solo bit realizado en *Octave*, usando software de libre distribución. El programa *SIDESIM-O* tiene la capacidad de leer un *script* que el usuario proporcione, reconocer cada una de las entradas, salidas y parámetros que contienen las funciones que se manden a llamar, para que, de esta manera, se escriban las funciones en el orden preciso de acuerdo al modulador  $\Sigma\Delta$  propuesto; además obtiene la respuesta en frecuencia y por último los parámetros SNR, SNDR y ENOB de la señal de salida del sistema. Todo con la mínima intervención del usuario, esto debido a que se creó un programa intérprete, ocasionando que el proceso de simulación de moduladores  $\Sigma\Delta$  sea transparente

para el usuario, ya que sólo tiene que escribir los bloques que requiere para el modelado de una arquitectura, sin escribir código alguno.

Debido a las diferencias encontradas en las métricas en la arquitectura de segundo orden se puede proponer un trabajo futuro: investigar cuáles son las razones que produzcan una diferencia en la estimación de los parámetros, simulando arquitecturas de tercer y cuarto orden, y a través de un análisis estadístico adecuado estudiar dicho comportamiento.

Con la ayuda de la herramienta computacional *SIDESIM-O* el usuario sólo tendrá que indicar el nombre del archivo, para que, de esta manera, comience el análisis. Es necesario destacar, que dicha herramienta puede ser mejorada, al incluir los diferentes tipos de limitantes no ideales que existen en el circuito real, para que de esta manera se le proporcione al usuario una información completa del desempeño del modulador cuando se implemente con elementos reales.

Además, el usuario podrá estudiar el comportamiento de una arquitectura de interés sin la adquisición de una licencia o licencias, es posible también que el usuario pueda modificar la herramienta desarrollada, por lo que también es adaptable, de fácil uso y de libre distribución.

## **6. Bibliografía y Referencias**

- [1] LM111 datasheet. <http://www.ti.com/cn/cn/lit/ds/symlink/lm311-n.pdf>. Junio 2016.
- [2] A. K. Sahu, V. K. Chandra, G. R. Sinha, "Improved SNR and ENOB of Sigma-Delta Modulator for Post Simulation and High Level Modeling of Built-in-Self-Test Scheme". *International Journal of Computer Applications*. 2014. Pp. 11-14.
- [3] A. Buhmann, M. Keller, M. Maurer, M. Ortmanns, Y. Manoli, DISCO - A toolbox for the discrete-time simulation of continuous-time sigma-delta modulators using MATLAB. 2007. Pp. 1082-1085.
- [4] N. Sirdeshpande, R. Desai, Simulink Based Approach for Modelling Of Third Order Sigma Delta Modulator For Audio Applications. 2015.

- [5] R. Mingyuan, "The Design of High Performance, High Resolution Two-Order Delta-Sigma Modulator". *Sensors & Transducers*. Vol. 22. No. 6. 2013. Pp. 155.
- [6] S. Jaykar, P. Palsodkar, P. Dakhole, "Modeling of Sigma-Delta Modulator Non-Idealities in MATLAB/SIMULINK". 2011 International Conference on Communication Systems and Network Technologies, 978-0-7695-4437-3/112011 IEEE DOI 10.1109/CSNT.2011.112
- [7] A. Dendouga, N. E. Bouguechal, S. Kouda, S. Barra, B. Lakehal, "Contribution to the modeling of a non-ideal Sigma-Delta modulator". *Journal of Computational Electronics*. Vol. 11. No. 4. 2012. Pp. 321-329.
- [8] M. Bashir, S. R. Patri, K. S. R. Krishnaprasad, "MATLAB/SIMULINK based time-domain behavioral modeling of multibit sigma-delta converters". In *2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT)*. March 2016. Pp. 132-136.
- [9] F. Wu, Z. J. Chen, M. Zhao, D. Y. Xu, G. C. Shen, W. G. Lu, Y. C. Zhang, "A new op-amp noise model for switched-capacitor sigma-delta modulator in SIMULINK". *IEEE International Conference On in Electron Devices and Solid-State Circuits (EDSSC)*. June 2014. Pp. 1-2.
- [10] L. Fucik, R. Vrba, "MATLAB Model of 16-bit Switched-capacitor Sigma delta Modulator with Two-step Quantization Process". *IEEE Transactions on Circuits and System*. 2006.

## 7. Autores

Ximena Victoria Bouzas Munguía es estudiante de Ing. en Electrónica en la UAM-Azcapotzalco, Ciudad de México, donde su trabajo de proyecto terminal es "Modelado en alto nivel de moduladores  $\Sigma\Delta$  usando *Octave*". Obtuvo el primer lugar en el área de Construcción de Prototipos en el XXII Congreso de Investigación CUAM-AcMor con el proyecto "Casa Telemática", Cuernavaca Morelos, en 2011. Actualmente es ayudante de investigación en el departamento de electrónica, en la Universidad Autónoma Metropolitana-Azcapotzalco



colaborando con el área de Instrumentación, procesamiento de señales y sistemas inteligentes. Sus intereses académicos incluyen el procesamiento digital de señales, así como el manejo, investigación e innovación de sistemas de control.

Javier Alducin Castillo recibió el título de Ing. en Electrónica por la UAM-Azcapotzalco, Ciudad de México, México, en 2008, el grado de M. en C. en Ingeniería Biomédica en 2014 por la UAM-Iztapalapa, Ciudad de México, México, obteniendo la Medalla al Mérito Universitario 2014. Actualmente es profesor en la Universidad Autónoma Metropolitana, unidad Azcapotzalco. Sus tópicos de investigación incluyen procesamiento digital de señales e imágenes, análisis de EEG y conectividad funcional cerebral.

Juan Jesús Ocampo Hidalgo es doctor en ingeniería por la universidad de tecnología de Darmstadt, en Darmstadt, Alemania. Trabajó como diseñador de circuitos integrados CMOS analógicos y de señal mixta en AnaFocus S de RL en Sevilla, España, donde desarrolló circuitos integrados para lectura y conversión A/D para pixeles activos en tecnologías CMOS nanométricas. Actualmente es profesor-investigador de tiempo completo en el Departamento de Electrónica de la Universidad Autónoma Metropolitana unidad Azcapotzalco. Sus intereses académicos son el diseño de circuitos integrados CMOS analógicos y de señal mixta.

Iván Vázquez Álvarez obtuvo su título de título de Ing. en Electrónica por la UAM-A, Ciudad de México, México, en 1999, el grado de M. en C. en Ingeniería Eléctrica en 2004 por el CINVESTAV-IPN en Guadalajara, Jalisco, México, y el grado de Dr. en el 2006 por la Universidad Estatal de Moscú – Lomonosov, Federación Rusa. Actualmente es profesor en la Universidad Autónoma Metropolitana. Sus tópicos de investigación incluyen el modelado y simulación de sistemas mecánicos, los sistemas no lineales y el control aplicado a sistemas automotrices.