

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

**INFORME DE MATERIA DE GRADUACIÓN**

**“PROCESAMIENTO DIGITAL DE SEÑALES”**

**“RESTAURACIÓN DE VOZ DETERIORADA POR RUIDO DE PULSOS”**

Previa a la obtención del Título de:

**INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**

Presentado por:

**JORGE ANDRÉS BRITO COLLANTES**

**MELISSA NATALY CORTEZ VEGA**

Profesor:

**ING. PATRICIA CHÁVEZ BURBANO**

**GUAYAQUIL – ECUADOR**

**AÑO**

**2009**

## AGRADECIMIENTOS

Primeramente agradezco a Dios por su presencia e inspiración desde el inicio hasta el final de este proyecto, a mi Papá y a mi Mamá por ser modelos de ética y responsabilidad, además de recibir su incondicional apoyo durante toda mi carrera.

A los Ingenieros que fueron clave en mi formación profesional. A la Ing. Patricia Chávez Burbano, por su consideración, asistencia, y paciencia durante la totalidad del proceso de la materia de graduación.

**Jorge Brito Collantes.**

Agradezco a Dios por la terminación de este trabajo, a mis padres por su confianza y su apoyo incondicional durante mi vida estudiantil.

A la Ing. Patricia Chávez, por su paciencia, dedicación y esfuerzo para la culminación de mi carrera.

**Melissa Cortez Vega.**

# DEDICATORIAS

A Dios,  
a la Madre Dolorosa.

A mi familia,  
y a mis amigos.

**Jorge Brito Collantes.**

A Dios,

A mi Padre, por ser mi ejemplo de esfuerzo y superación.

A mi madre, por estar siempre conmigo, apoyándome ante todas las adversidades.

A mi hermana, a mis familiares y amigos que con cariño ayudaron a culminar mi carrera profesional.

**Melissa Cortez Vega.**

## TRIBUNAL DE SUSTENTACIÓN

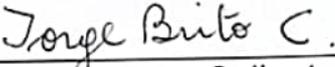
  
Ing. Juan Carlos Avilés  
DELEGADO DECANO FIEC

  
Ing. Patricia Chávez  
PROFESORA MATERIA DE  
GRADUACION

## DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Tesis de Grado, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITECNICA DEL LITORAL”

(Reglamento de Graduación de la ESPOL)

  
\_\_\_\_\_  
Jorge Brito Collantes.

  
\_\_\_\_\_  
Melissa Cortez Vega.

## **RESUMEN**

Se quiere presentar la implementación de un método fácil y eficiente adaptativo basado en una predicción lineal y filtración para la cancelación de un ruido de pulso y obtener una considerable mejoría en la calidad de la señal, el método a aplicar para resolver estos problemas, que es generalmente usado para el procesamiento de imágenes, es el de filtro de mediana (median filter).

Se planea ingresar al sistema una señal (voz humana) afectada por el ruido de pulso, y mediante la utilización de filtros mejorarla de tal forma que la salida sea una señal entendible.

# ÍNDICE GENERAL

RESUMEN

ÍNDICE GENERAL

ABREVIATURAS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN .....	1
1. METODOLOGÍA Y MARCO TEÓRICO .....	3
2. IMPLEMENTACIÓN .....	7
2.1. Descripción del Software .....	7
2.2. Requerimientos de Hardware y Software .....	8
2.3. Funcionamiento.....	8
2.3.1. Adquisición de archivo de audio .....	9
2.3.2. Acondicionamiento de la señal .....	11
2.3.3. Bloque de Filtrado .....	14
2.3.4. Bloque Detector de Ruido .....	18
2.3.5. Bloque Sostenedor .....	26

2.3.6. Bloque Selector .....	29
2.3.7. Conversión de archivo de audio .....	33
3. ANÁLISIS DE RESULTADOS .....	35

CONCLUSIONES

RECOMENDACIONES

ANEXOS

ANEXO A: Encuestas

ANEXO B: Manual de Usuario

REFERENCIAS

## ABREVIATURAS

MF: Median Filter.

s. Segundos

## ÍNDICE DE FIGURAS

Figura 1.1.	Muestras afectadas por ruido .....	3
Figura 1.2.	Diagrama de Bloques general .....	4
Figura 2.1.	Interfaz Gráfica de Simulink.....	8
Figura 2.2.	Esquema de implementación .....	9
Figura 2.3.	Señal de audio .....	10
Figura 2.4.	Ruido de pulso .....	10
Figura 2.5.	Múltiples pulsos .....	11
Figura 2.6.	Bloques “From Wave File” y “Unbuffer” .....	12
Figura 2.7.	Muestra de 2 canales estéreo, a un rango de tiempo de 17s. ...	12
Figura 2.8.	Muestra de 2 canales estéreo, a un rango de tiempo de 0.1s ..	13
Figura 2.9.	Muestra de 2 canales estéreo, a un rango de tiempo de $9 \times 10^{-3}$ s .....	13
Figura 2.10.	Bloque de Filtrado.....	14
Figura 2.11.	Señal de salida Out1 del Bloque de Filtrado a un rango de tiempo de 17s .....	15
Figura 2.12.	Señal de salida Out1 del Bloque de Filtrado a un rango de tiempo de 1s .....	16
Figura 2.13.	Señal de salida Out1 del Bloque de Filtrado a un rango de tiempo de 0.1s .....	16
Figura 2.14.	Señal de salida Out2 del Bloque de Filtrado a un rango de tiempo de 17s .....	17
Figura 2.15.	Señal de salida Out2 del Bloque de Filtrado a un rango de tiempo de 1s .....	17
Figura 2.16.	Señal de salida Out3 del Bloque de Filtrado a un rango de tiempo de 0.1s .....	18
Figura 2.17.	Bloque detector de ruido.....	18
Figura 2.18.	Primera derivada de la señal a un rango de tiempo de 17s.....	19
Figura 2.19.	Primera derivada de la señal a un rango de tiempo de 0.1s.....	20
Figura 2.20.	Segunda derivada de la señal a un rango de tiempo de 17s.....	20
Figura 2.21.	Segunda derivada de la señal a un rango de tiempo de 0.1s.....	21
Figura 2.22.	Señal elevada al cuadrado a un rango de tiempo de 17s.....	22
Figura 2.23.	Señal elevada al cuadrado a un rango de tiempo de $9 \times 10^{-3}$ s.....	22
Figura 2.24.	Salida del selector de canal a un rango de tiempo de 17s.....	23
Figura 2.25.	Salida del selector de canal a un rango de tiempo de $9 \times 10^{-3}$ s.....	23
Figura 2.26.	Señal con referencia c a un rango de tiempo de 17s.....	24
Figura 2.27.	Señal con referencia c a un rango de tiempo de 1s.....	25
Figura 2.28.	Señal de salida Out3 del Bloque Detector de Ruido a un rango de tiempo de 17s .....	25
Figura 2.29.	Señal de salida Out3 del Bloque Detector de Ruido a un rango de tiempo de $7 \times 10^{-3}$ s.....	26
Figura 2.30.	Bloque Sostenedor .....	27
Figura 2.31.	Señal de salida Out4 del Bloque Sostenedor a un rango de tiempo de 17s.....	28

Figura 2.32. Señal de salida Out4 del Bloque Sostenedor a un rango de tiempo de 0.03s.....	28
Figura 2.33. Bloque Selector .....	29
Figura 2.34. Detección de señal con ruido a un rango de tiempo de 17s .....	30
Figura 2.35. Detección de señal con ruido a un rango de tiempo de 1s .....	31
Figura 2.36. Detección de señal con ruido a un rango de tiempo de 0.1s .....	31
Figura 2.37. Salida del Bloque Selector a un rango de tiempo de 17s .....	32
Figura 2.38. Salida del Bloque Selector a un rango de tiempo de 1s .....	32
Figura 2.39. Salida del Bloque Selector a un rango de tiempo de 0.1s .....	33
Figura 2.40. Bloques “Buffer” y “To Wave File” .....	33
Figura 3.1. Comparación entre señal de entrada y salida del sistema en 17s .....	35
Figura 3.2. Comparación entre señal de entrada y salida del sistema en 0.1s .....	36
Figura 3.3. Muestra con un ancho de pulso de 0.003 en 0.1s .....	37
Figura 3.4. Muestra con un ancho de pulso de 0.009 en 0.1s .....	38
Figura 3.5. Muestra con un ancho de pulso de 0.00625 en 0.1s .....	39
Figura 3.6. Señal con ruido, calidad estéreo .....	41
Figura 3.7. Señal de salida, calidad mono .....	42

## ÍNDICE DE TABLAS

Tabla I.	Tabla de Formato de encuesta.....	47
Tabla II.	Tabla de Porcentajes de la encuesta .....	48

## INTRODUCCIÓN

El ruido en el procesamiento de señales es cualquier señal indeseada, que se introduce en los datos a través de cualquier sistema eléctrico utilizado para almacenamiento, transmisión y/o procesamiento.

Cuando encontramos una imagen o un archivo de audio afectado por ruido, se desea mejorar su presentación para una aplicación específica.

La reducción del ruido es el proceso de eliminación de un porcentaje considerable del ruido de la señal; las técnicas de reducción de ruido son muy parecidas independientemente del tipo de señal que está siendo procesada. Sin embargo la implementación de estas técnicas son las que varían dependiendo del tipo de señal.

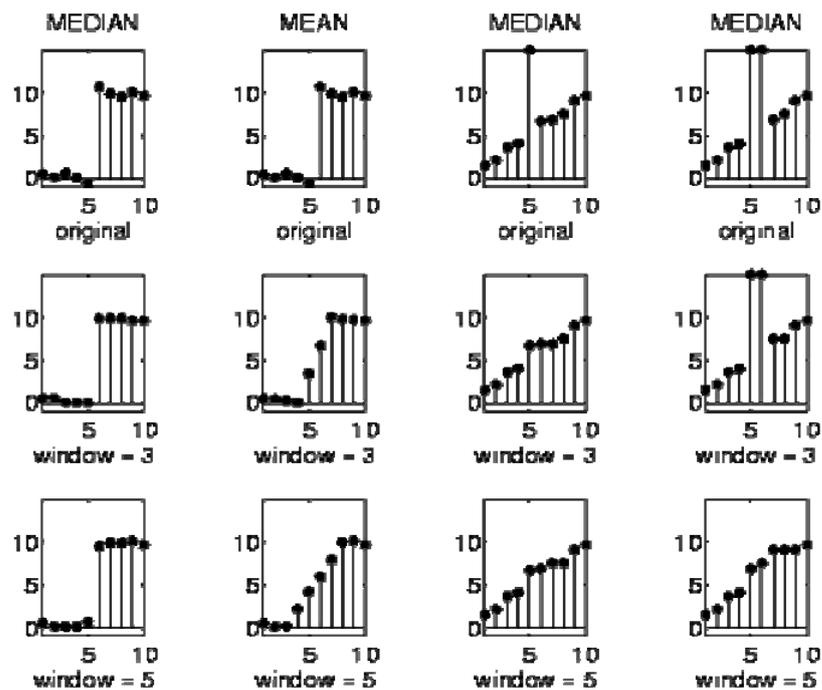
Todos los dispositivos de grabación digital o analógica tienen rasgos que los hacen susceptibles al ruido; cuando se utiliza la tecnología de grabación analógica, grabaciones de sonido presentan un tipo de ruido conocido como siseo de cinta; además las señales de audio en un rango amplio de aplicaciones están sujetas a ruidos de pulso, ya sea por el rayado de un disco LP, mala sincronización en grabaciones digitales, o los efectos de propagación de los sistemas móviles [1]; la deterioración de la señal de audio puede ser severa.

Varios Software para la cancelación de este tipo de ruido han sido desarrollados, generalmente estos funcionan bien en señales de audio severamente afectadas pero debido a los complejos algoritmos que utilizan, el tiempo de procesamiento de la señal es lento.

# CAPÍTULO I

## METODOLOGÍA Y MARCO TEÓRICO

La supresión del ruido de pulsos se realizará a través de un Median Filter (MF). El MF es un filtro no lineal que utiliza una técnica para la suavización de la señal y supresión del ruido de pulsos. En el caso de una dimensión se compone de una ventana deslizante de un número de elementos impares a través de la señal, reemplazando la muestra del centro por la mediana de las muestras en la ventana [2].



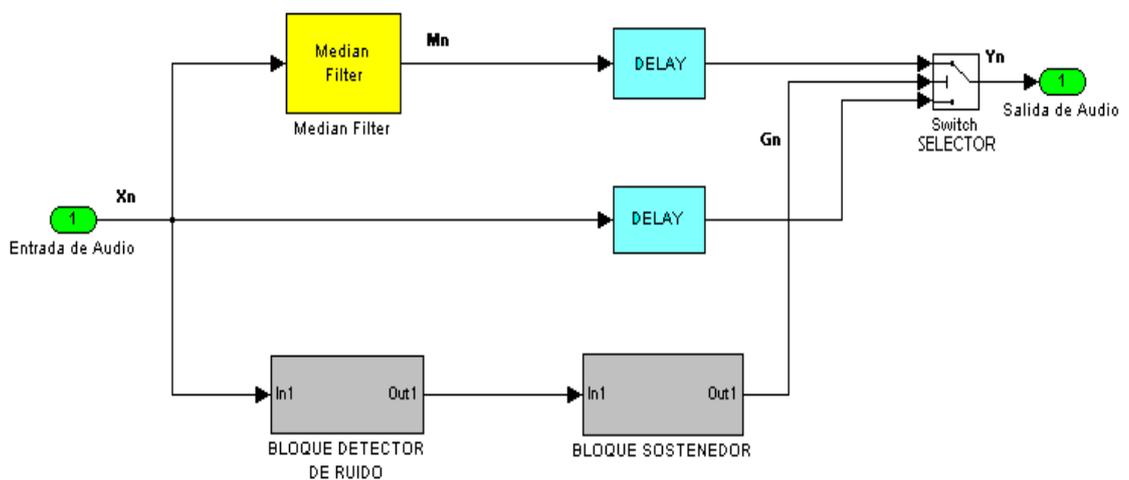
**Figura 1.1** Muestras afectadas por ruido

La figura 1.1 muestra el uso de ventanas de tamaños de 3 y 5 muestras respectivamente.

Las dos primeras columnas muestran una función escalón afectada por un ruido aleatorio; las dos últimas columnas muestran una función de línea recta con ruido además una y dos muestras que son considerablemente diferentes de las muestras vecinas.

Observamos que el MF en la primera columna conserva bien los bordes, el método de la segunda columna equivaldría al de un filtro pasa bajos que suaviza los bordes completamente. Las columnas 3 y 4 muestran la importancia del tamaño de la ventana; una muestra fuera de rango puede ser fácilmente quitada con un tamaño de ventana de 3, mientras que 2 muestras que están fuera de rango solo pueden ser quitadas con una ventana de mayor tamaño.

Se quiere presentar la implementación de un método adaptativo fácil y eficiente basado en una predicción lineal y filtrado [3] para la cancelación de un ruido de pulso y obtener una considerable mejoría en la calidad de la señal.



**Figura 1.2** Diagrama de Bloques general

La figura 1.2 muestra la entrada de audio afectada por un ruido de pulsos, la misma que pasará por un proceso que suprimirá el ruido de la señal.

El MF es deshabilitado salvo en las regiones donde los pulsos han sido detectados.

$$y_n = \begin{cases} m_n & \text{if } g_n > 0 \\ x_n & \text{otherwise} \end{cases}$$

Donde:

$$m_n = \text{med}\{x_{n-k}, \dots, x_n, \dots, x_{n+k}\}$$

Es el valor de mediana a través de una ventana de longitud  $N=2k+1$  muestras  $x_n$  y  $y_n$  son la entrada y secuencia de salida respectivamente,  $g_n$  es una señal de conmutación que se proporciona en el momento correcto para que el MF se active cuando existan pulsos. Es conocido que los MFs suprimen los pulsos en

anchos menores a  $k = \frac{N-1}{2}$  [4].

La señal  $g_n$  es generada en la parte inferior del diagrama de bloques donde la detección de pulsos y la señal de conmutación se producen. La discriminación de los pulsos y la conmutación es la base para el desarrollo de este modelo.

Para una detección de pulso más sólida de la señal de entrada, se obtiene la segunda derivada discreta para que estos pulsos del ruido de la señal aumenten en amplitud.

$$D^2 \{x_n\} = x_{n-1} - 2x_n + x_{n+1} \quad [5]$$

Este aumento de amplitud es considerable a la salida de este proceso.

Debido a las características oscilatorias del ruido de pulsos el MF debe mantenerse activo durante la totalidad de las componentes del ruido de la señal. La señal de conmutación debe coincidir con la forma de onda del ruido.

## CAPÍTULO II

### IMPLEMENTACIÓN

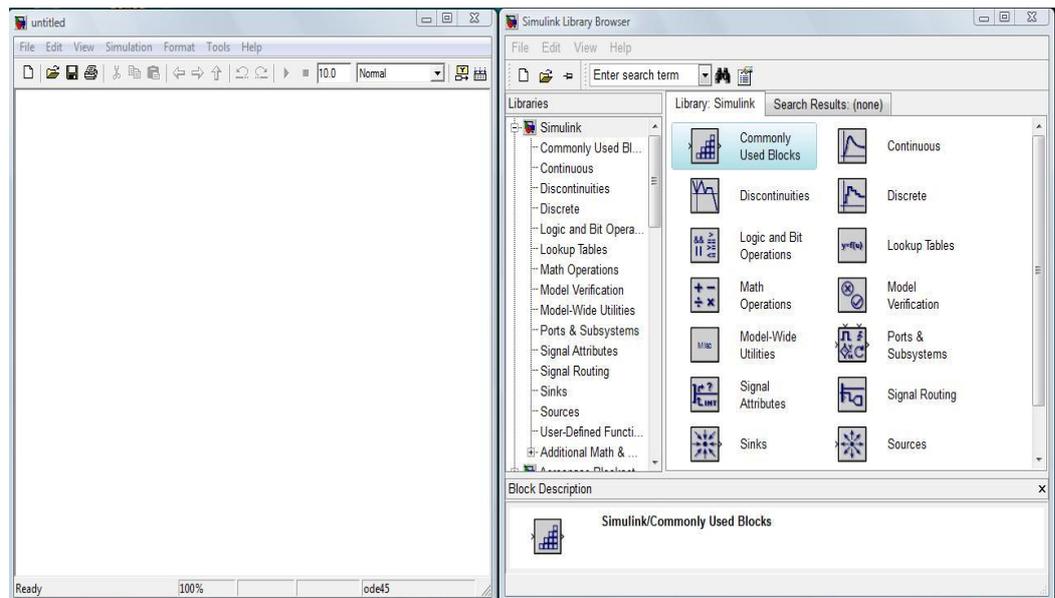
#### 2.1. DESCRIPCIÓN DEL SOFTWARE

**MATLAB** (abreviatura de *MATrix LABoratory*, "laboratorio de matrices") es un software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows y Apple Mac OS X.

El paquete MATLAB 7.7.0 (R2008b), dispone de dos herramientas adicionales que expanden sus prestaciones, a saber, **Simulink** (plataforma de simulación multidominio) y **Guide** (editor de interfaces de usuario - GUI).

**Simulink**, desarrollado por The MathWorks, es una plataforma comercial para modelar, simular y analizar sistemas dinámicos multidominio el cual posee un juego de herramientas incorporado de gran alcance en el tratamiento de señales.

Su interfaz gráfica es una herramienta de diagramas de bloques personalizables y un conjunto de bibliotecas de bloques. Que ofrece una estrecha integración con el resto del entorno de MATLAB.



**Figura 2.1** Interfaz Gráfica de **Simulink**

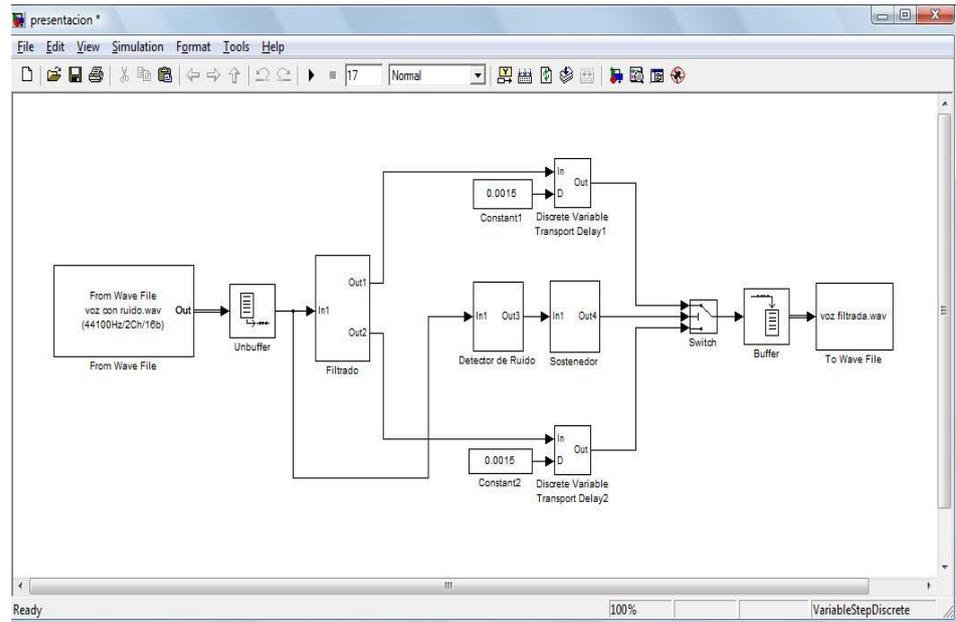
## 2.2. REQUERIMIENTOS DE HARDWARE Y SOFTWARE

Para un óptimo desempeño de la simulación son necesarios los siguientes requerimientos de hardware y software mínimos:

- Desktop: Procesador Dual Core 1.73 GHz”.
- Memoria RAM 2GHz
- Sistema Operativo Windows XP Service Pack 2
- Disponibilidad de puertos USB.

## 2.3. FUNCIONAMIENTO

Ya que **Simulink** se utiliza ampliamente en la teoría de control y procesamiento de señales digitales para simulación y diseño, se utilizó su interfaz gráfica como herramienta de trabajo para desarrollar el proyecto como se puede apreciar en la figura 2.2.



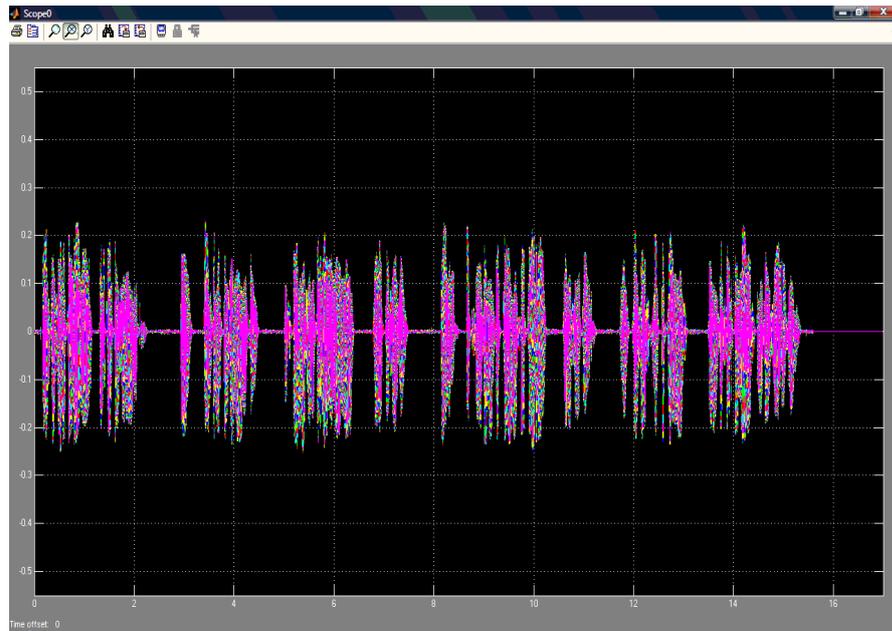
**Figura 2.2** Esquema de implementación.

### 2.3.1. Adquisición de archivo de audio

El archivo de sonido usado es una señal de audio (voz) con una duración de 15 segundos con una frecuencia de muestra de datos de 44100Hz.

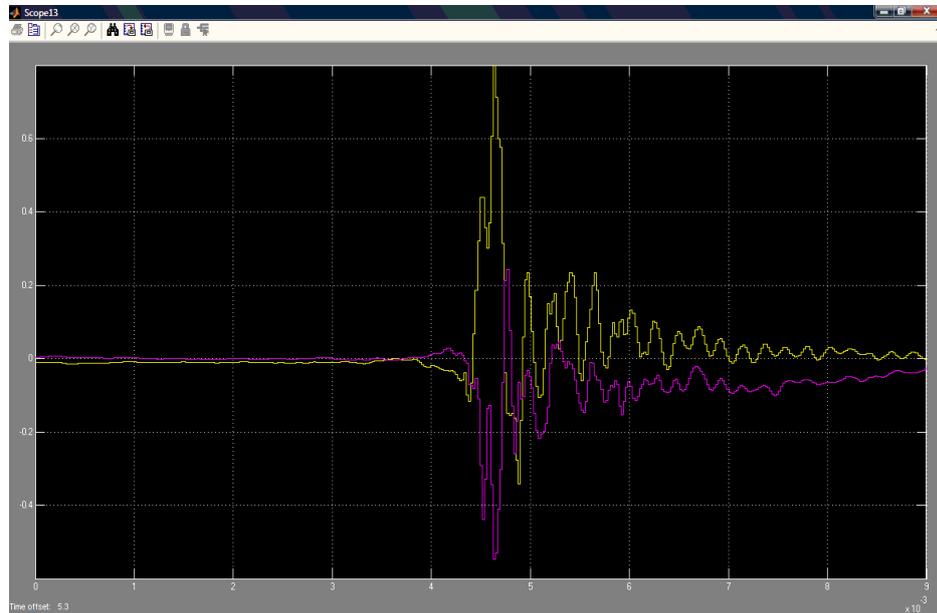
El ruido que afectara a la señal de audio, es un ruido de pulsos (clicks y pops) con una frecuencia de muestra de datos de 44100Hz y está conformado por 11 pulsos que se repiten periódicamente.

En la figura 2.3 se puede observar la señal de audio real no afectada por ruido.



**Figura 2.3** Señal de audio

En las figuras 2.4 y 2.5 se puede observar el ruido de pulsos que afectará la señal de entrada.



**Figura 2.4** Ruido de pulso.

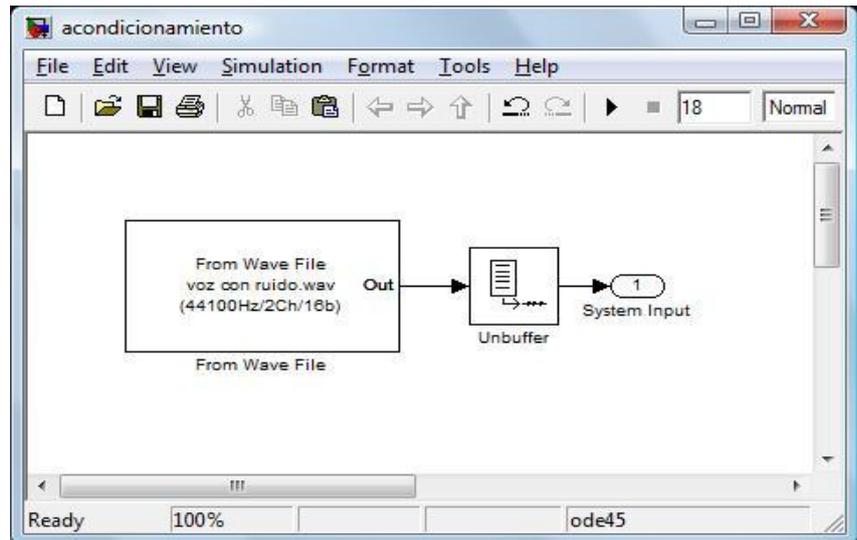


**Figura 2.5** Múltiples pulsos

Finalmente la señal de entrada al sistema es un archivo de audio real formado por la combinación de voz más un ruido de pulsos que provoca una deterioración en “voz con ruido.wav”; que tiene formato Windows PCM estéreo de 2 canales con un tamaño de 2.62 MB, velocidad de transmisión de 1411 Kbps a una frecuencia de muestra de datos de 44100Hz

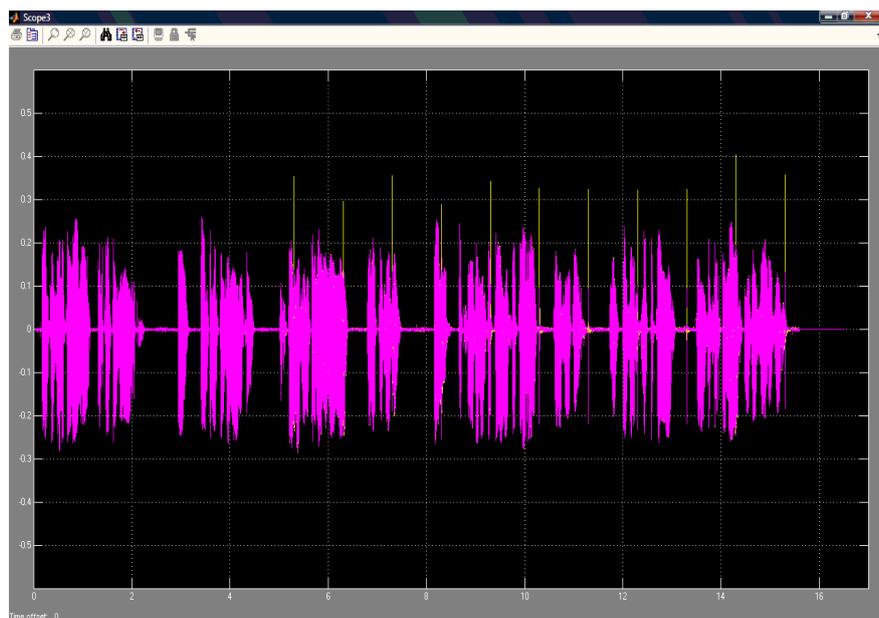
### **2.3.2. Acondicionamiento de la señal.**

El bloque encargado de adaptar el archivo de audio a la interface de Simulink es “From Wave File” el cual lo convierte en un vector de tramas de 256x2. Ver figura 2.6.



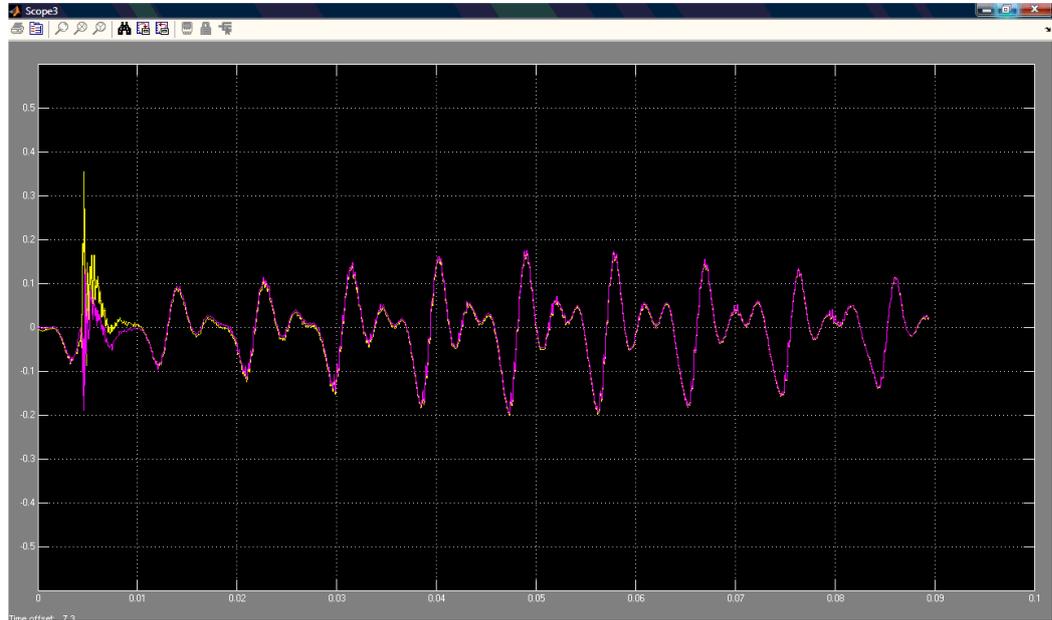
**Figura 2.6** Bloques “From Wave File” y “Unbuffer”

El vector ingresa al bloque “Unbuffer” que realiza la transformación de la matriz basada en tramas de  $256 \times 2$  a una salida basada en muestras; en este caso dos señales de  $1 \times 256$  que representan los 2 canales estéreo del archivo de audio de entrada como se puede apreciar en la figura 2.7.

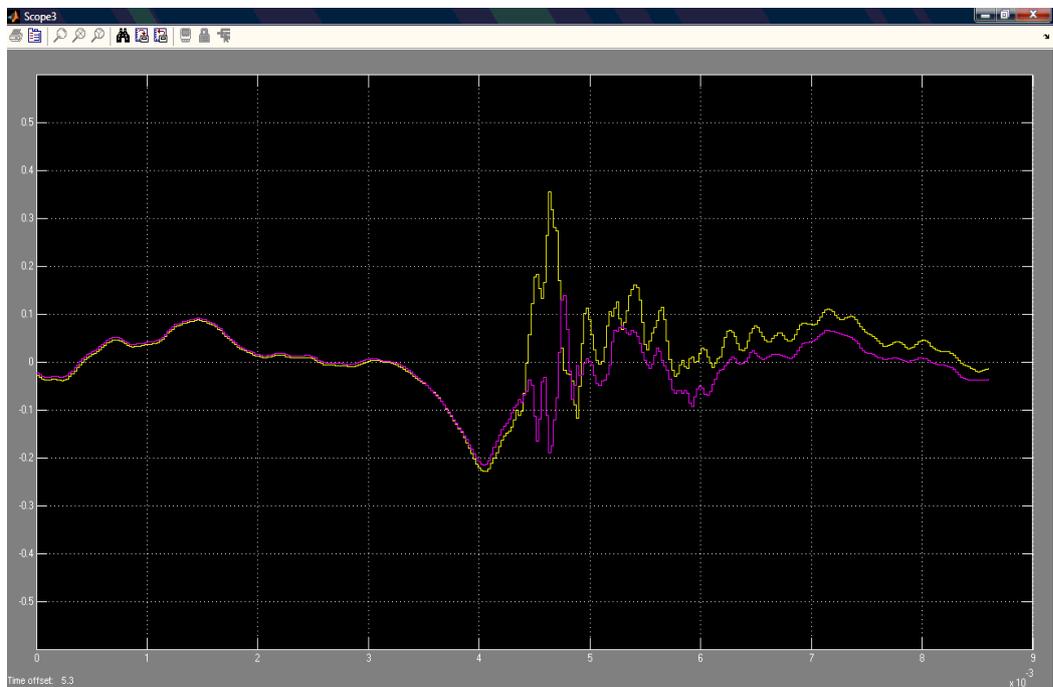


**Figura 2.7** Muestra de 2 canales estéreo, a un rango de tiempo de 17s.

En la figura 2.8 y 2.9 se puede observar los 2 canales estéreo de la señal, que se pueden diferenciar mejor por la variación de colores.



**Figura 2.8** Muestra de 2 canales estéreo, a un rango de tiempo de 0.1s.

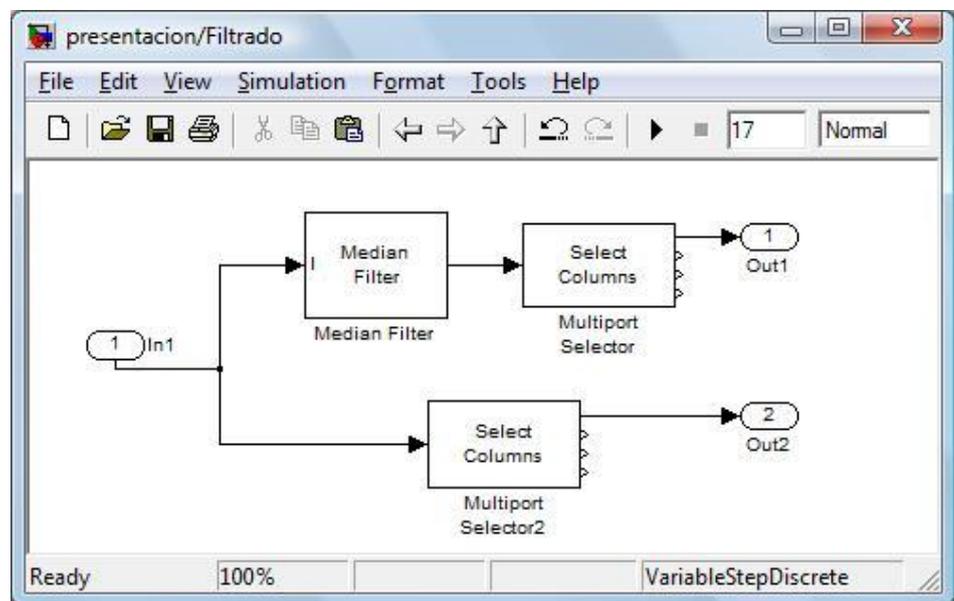


**Figura 2.9** Muestra de 2 canales estéreo, a un rango de tiempo de  $9 \times 10^{-3}$ s.

### 2.3.3. Bloque de Filtrado

Este bloque tiene la finalidad de obtener 2 señales, una filtrada (Out1) y otra sin filtrar (Out2) estas serán usadas posteriormente en el bloque selector.

La señal de entrada de audio (In1), estéreo de 2 canales se bifurca, como se puede apreciar en la figura 2.10.



**Figura 2.10** Bloque de Filtrado

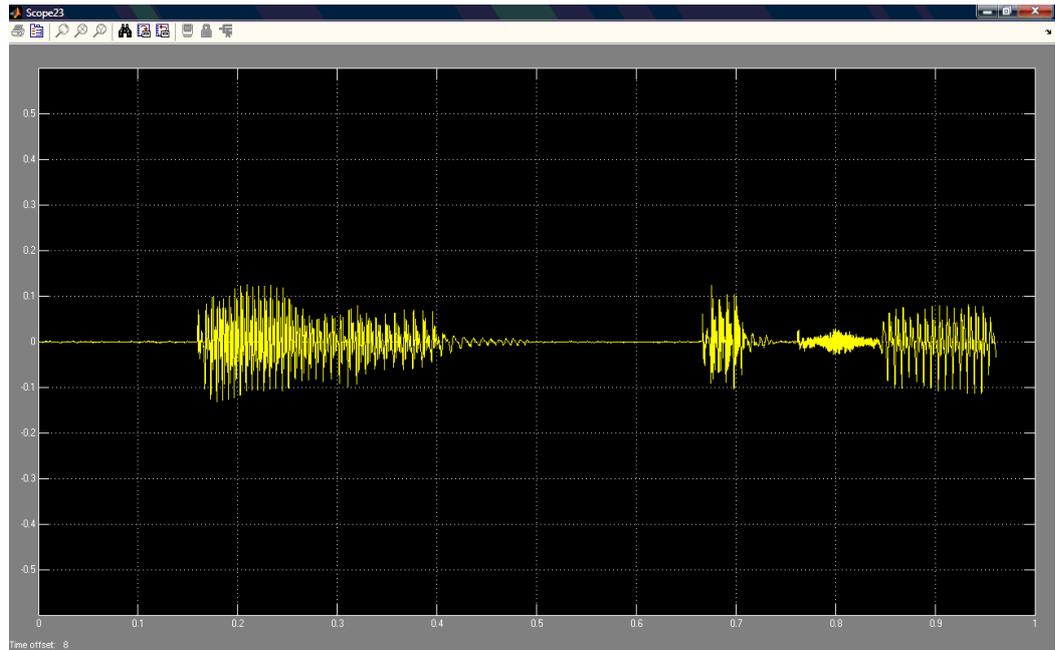
Por un camino la señal entra al bloque “Median Filter” donde es filtrada en su totalidad, en consecuencia de esto se produce una disminución significativa del ruido de pulsos además de una suavización en detalles de frecuencia y pérdidas de amplitud de la misma.

Esta señal filtrada entra al bloque “Multiport Selector” el cual selecciona uno de los dos canales estéreo para trabajar con dicha señal.

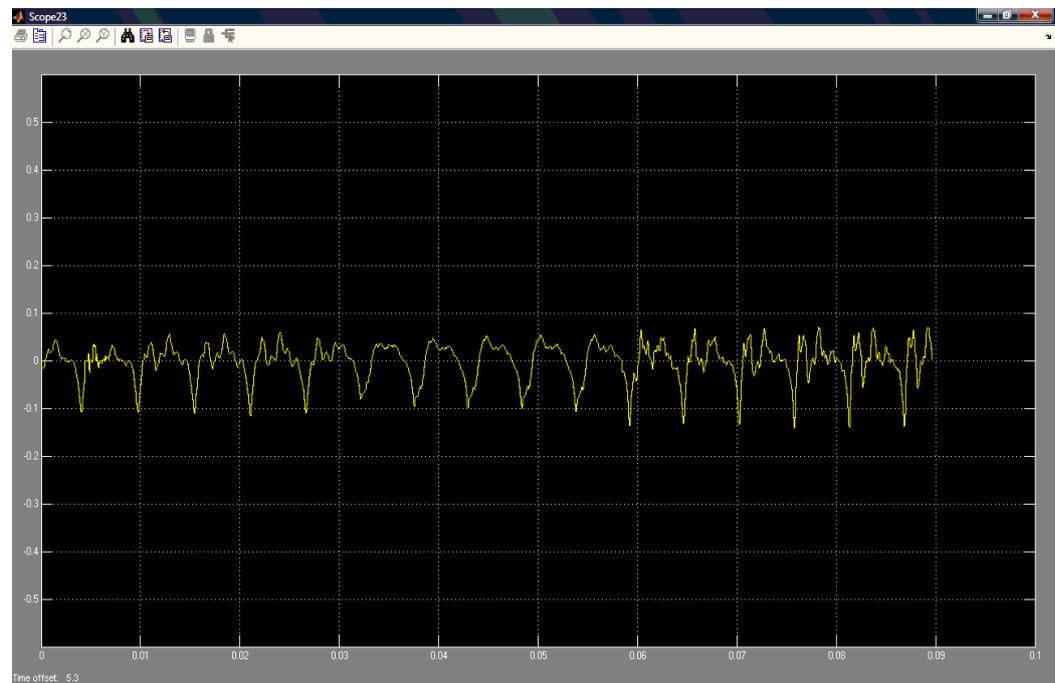
En la figura 2.11, 2.12 y 2.13 se puede observar la señal de audio luego de pasar por el MF y el selector de canales.



**Figura 2.11** Señal de salida Out1 del Bloque de Filtrado a un rango de tiempo de 17s.



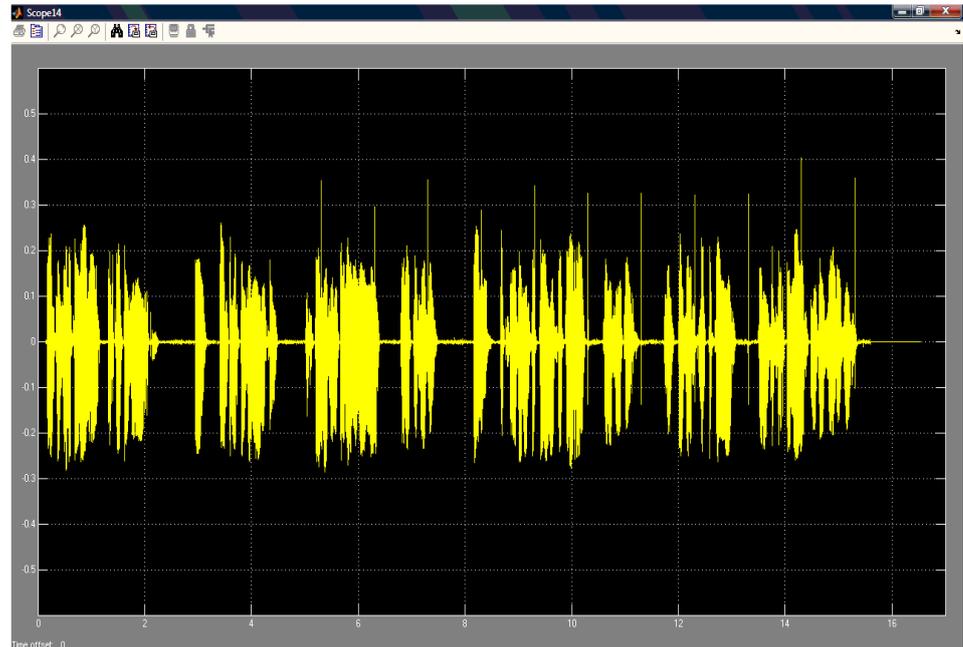
**Figura 2.12** Señal de salida Out1 del Bloque de Filtrado a un rango de tiempo de 1s.



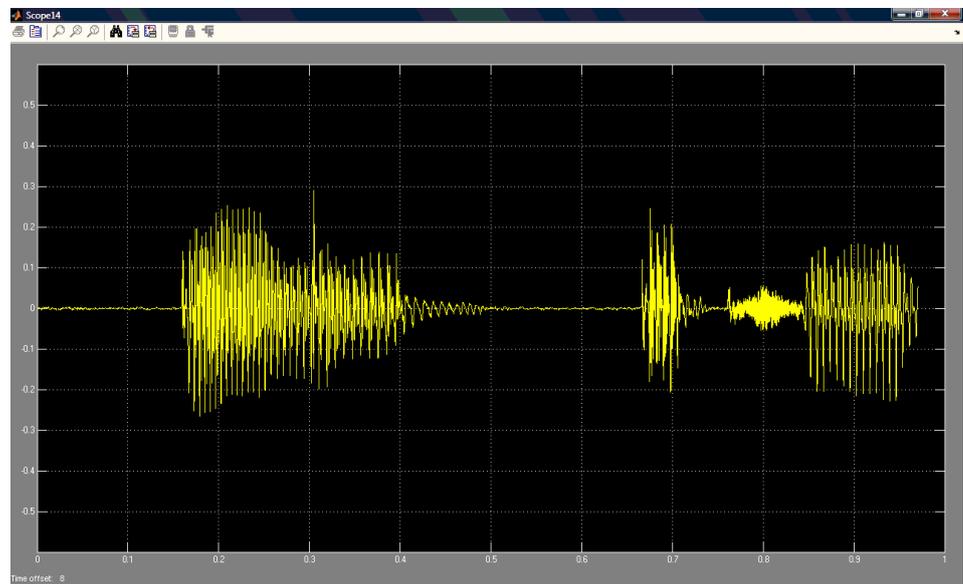
**Figura 2.13** Señal de salida Out1 del Bloque de Filtrado a un rango de tiempo de 0.1s.

En el segundo camino que toma la señal; se realiza otra selección de canal, en este caso de la señal no filtrada, a través del bloque

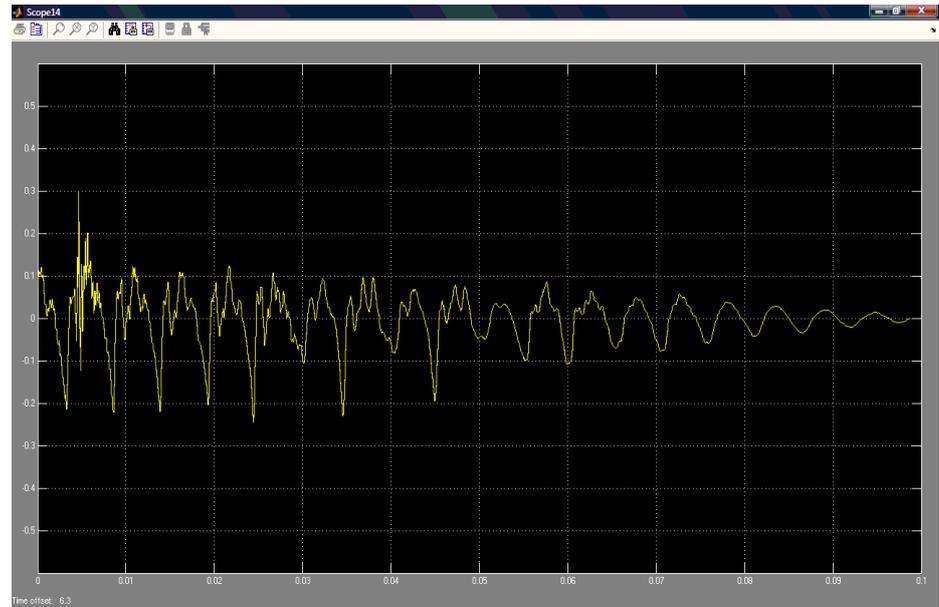
“Multiport Selector”. La señal de salida de este bloque se aprecia en la figura 2.14.



**Figura 2.14** Señal de salida Out2 del Bloque de Filtrado a un rango de tiempo de 17s.



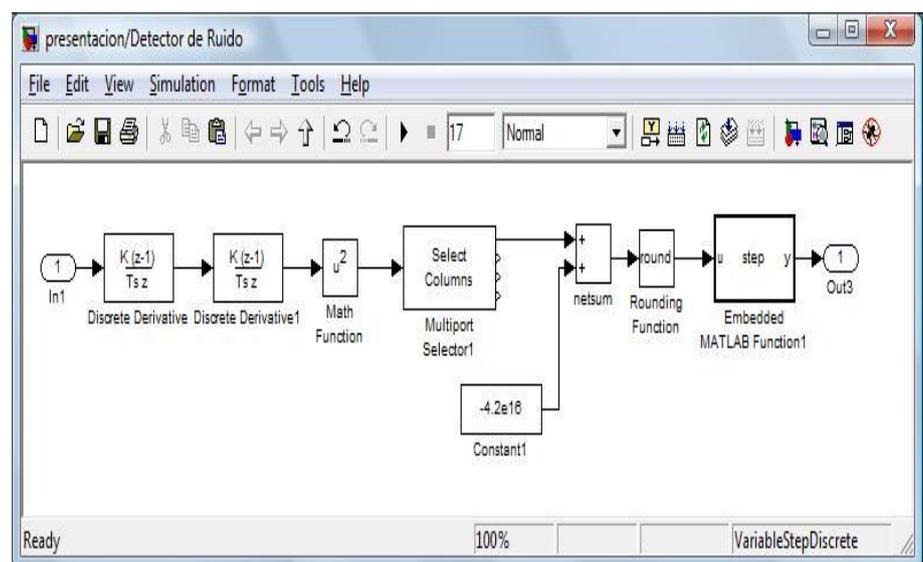
**Figura 2.15** Señal de salida Out2 del Bloque de Filtrado a un rango de tiempo de 1s.



**Figura 2.16** Señal de salida Out2 del Bloque de Filtrado a un rango de tiempo de 0.1s.

### 2.3.4. Bloque Detector de Ruido

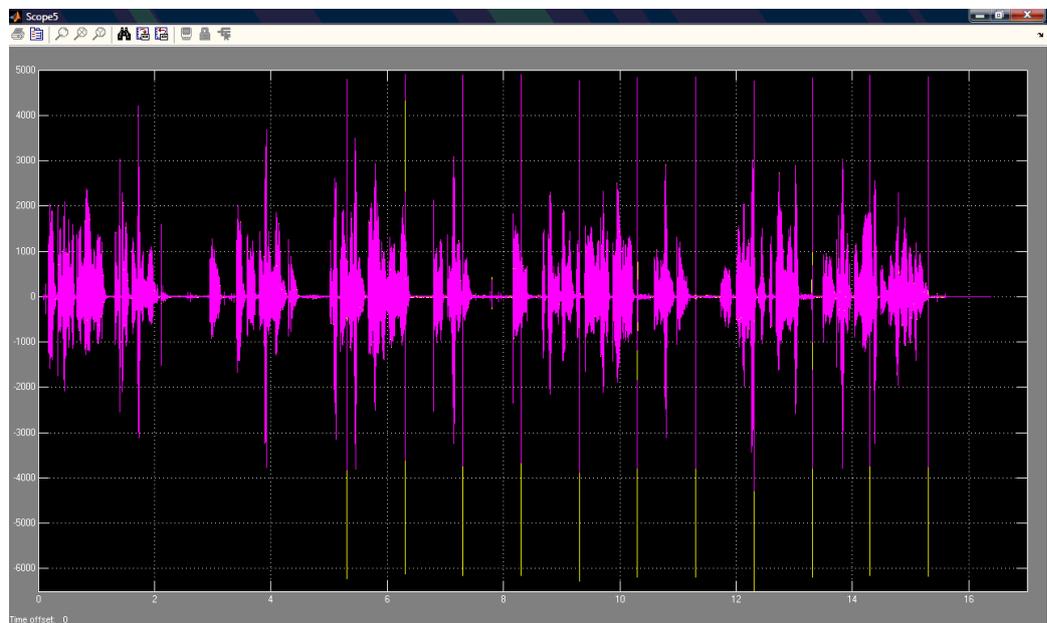
La finalidad de este bloque es la obtención de una señal de pulsos de amplitud uno (Out3) para la identificación o ubicación del ruido. Esta señal será usada posteriormente en el bloque selector.



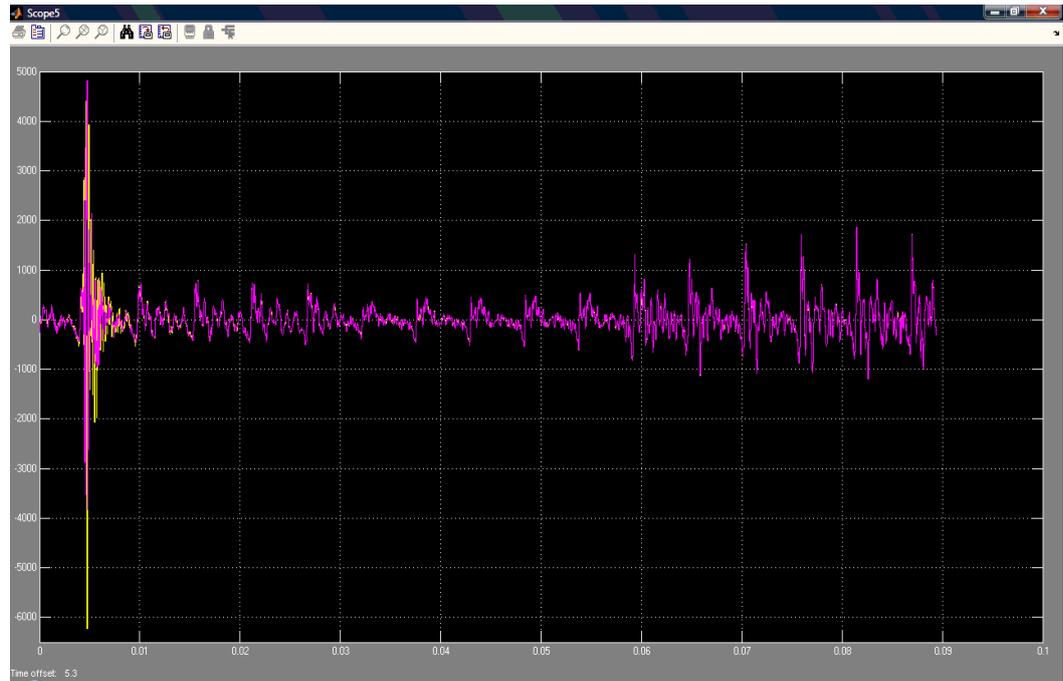
**Figura 2.17** Bloque Detector de Ruido

La señal estéreo de 2 canales pasa por dos bloques derivativos “Discrete Derivative” para obtener la segunda derivada de dicha señal para obtener un aumento de amplitud de las componentes de ruido de la señal.

En las figuras 2.18 y 2.19 se puede observar la señal obtenida por el primer bloque “Discrete Derivative” (primera derivada).



**Figura 2.18** Primera derivada de la señal a un rango de tiempo de 17s.

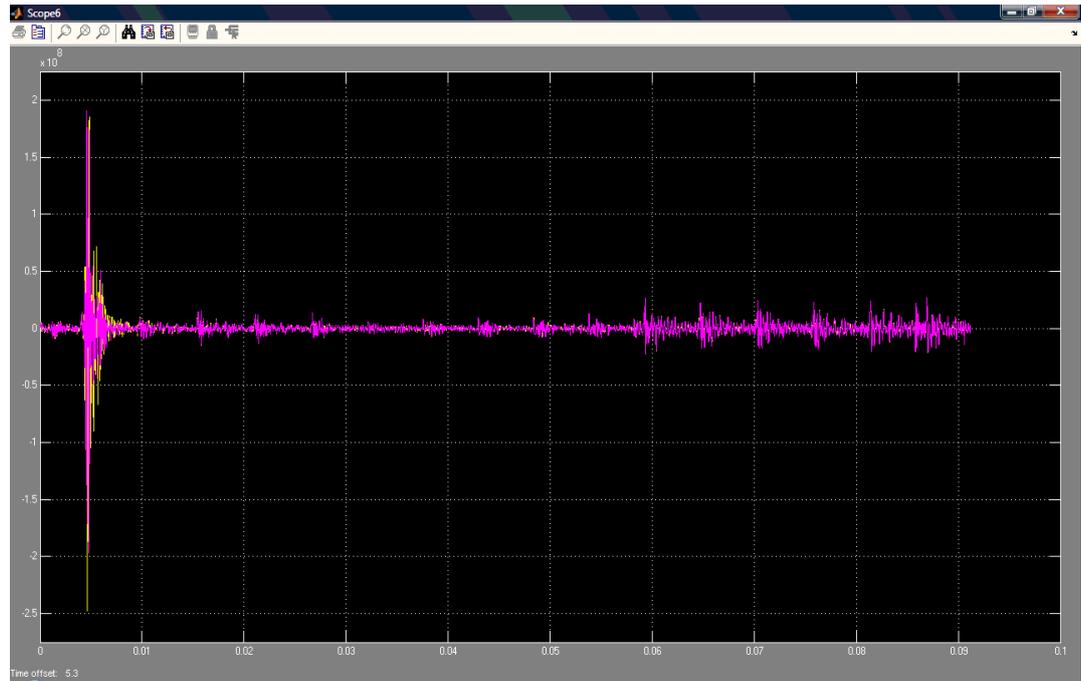


**Figura 2.19** Primera derivada de la señal a un rango de tiempo de 0.1s.

En la figura 2.20 y 2.21 se puede observar la señal obtenida después del segundo bloque “Discrete Derivative” (segunda derivada).

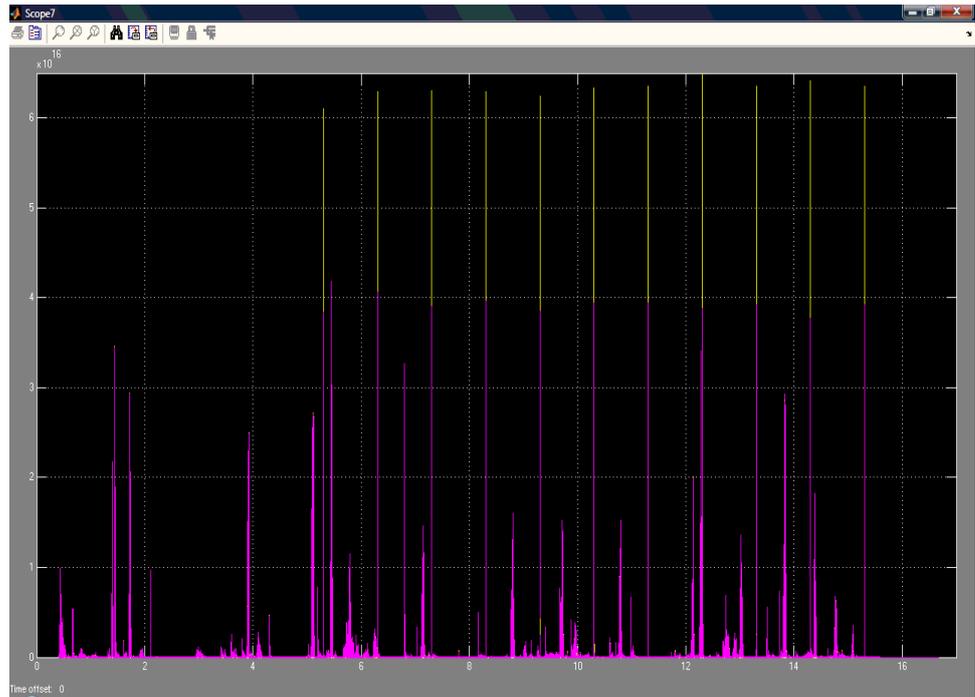


**Figura 2.20** Segunda derivada de la señal a un rango de tiempo de 17s

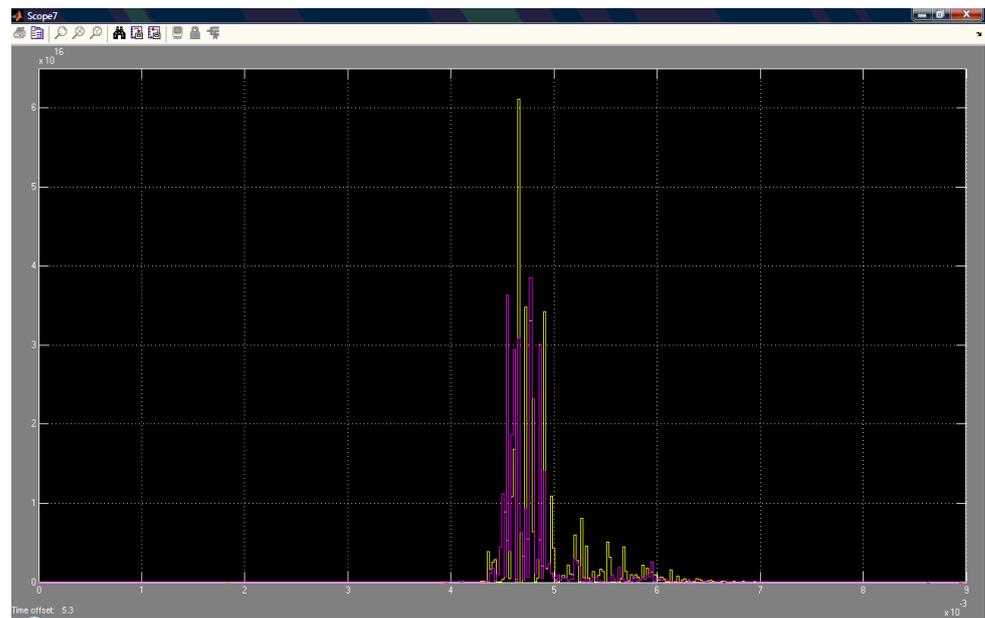


**Figura 2.21** Segunda derivada de la señal a un rango de tiempo de 0.1s.

Esta segunda derivada de la señal entra a un bloque “Math Function” que la eleva al cuadrado eliminando las componentes negativas de la misma; se obtiene además un mayor énfasis en los elementos del ruido como se puede observar en las figuras 2.22 y 2.23.

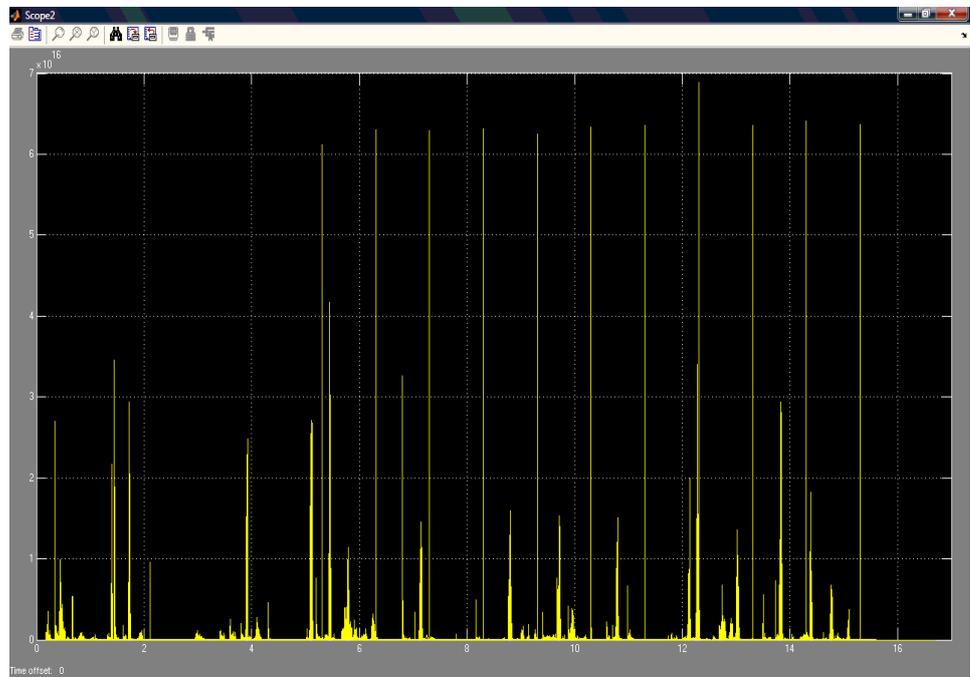


**Figura 2.22** Señal elevada al cuadrado a un rango de tiempo de 17s.

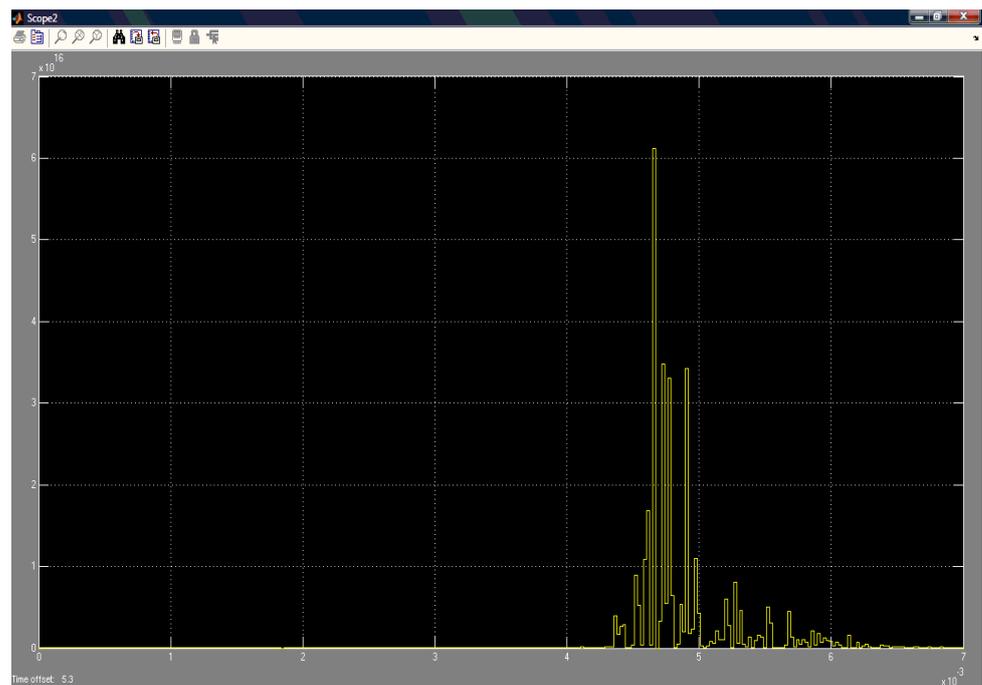


**Figura 2.23** Señal elevada al cuadrado a un rango de tiempo de  $9 \times 10^{-3}$ s.

Una vez que la señal es elevada al cuadrado entra al bloque “Multiport Selector” el cual selecciona uno de los dos canales de estéreo de la misma. Ver figuras 2.24 y 2.25.



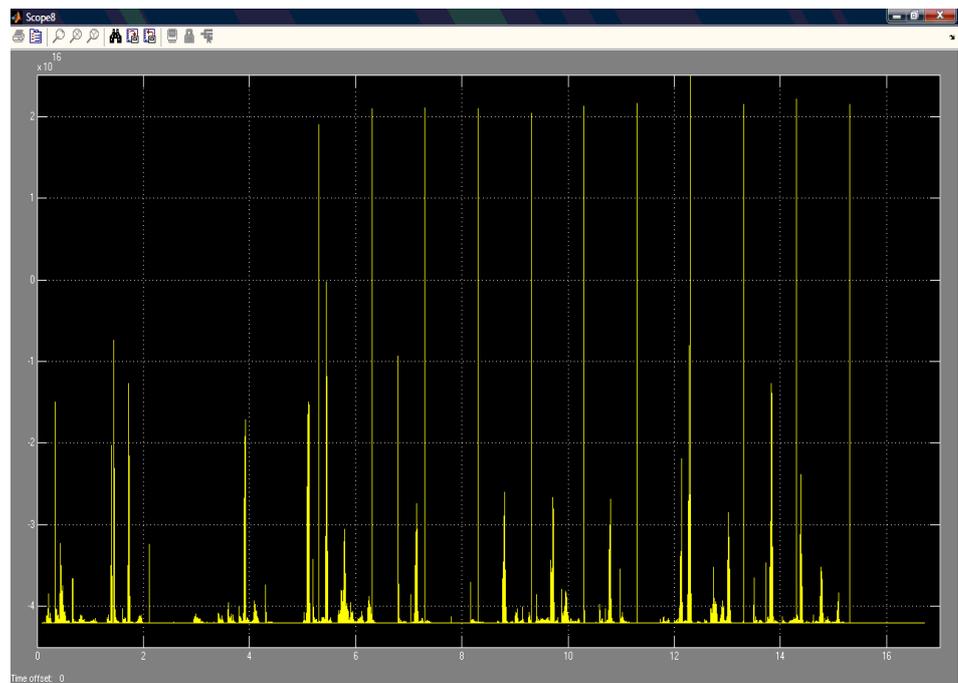
**Figura 2.24** Salida del selector de canal a un rango de tiempo de 17s.



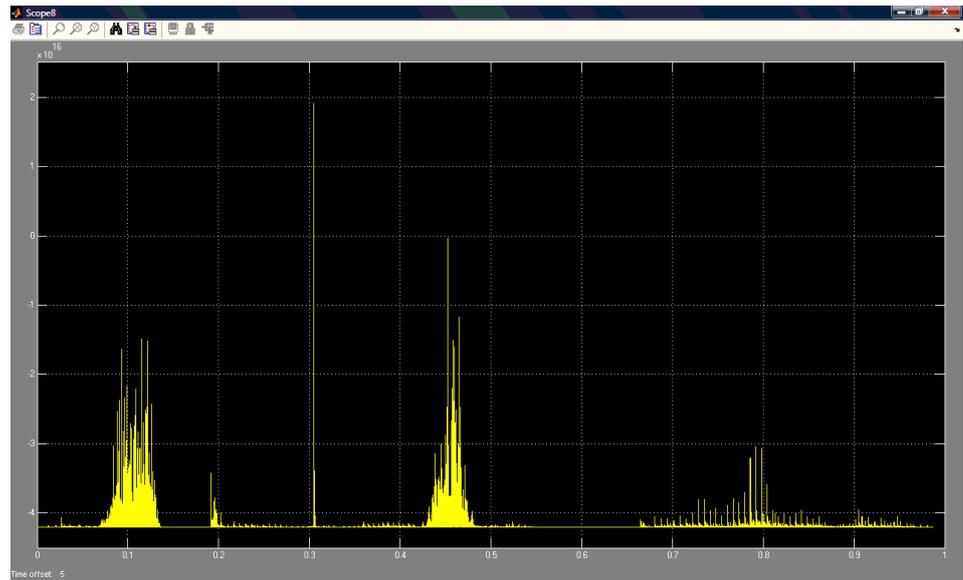
**Figura 2.25** Salida del selector de canal a un rango de tiempo de  $9 \times 10^{-3}$ s.

El canal seleccionado de la señal y una constante ( $c = -4.2 \times 10^{16}$ ) entran al bloque “netsum” y realizan la operación matemática de resta entre ellos, con la finalidad que la referencia con respecto a cero de la señal cambie; de esta manera solo los picos del ruido tendrán un valor mayor a cero.

Posteriormente la señal es redondeada por el bloque “Rounding Function”, como se observa en las figuras 2.26 y 2.27.



**Figura 2.26** Señal con referencia  $c$  a un rango de tiempo de 17s.



**Figura 2.27** Señal con referencia  $c$  a un rango de tiempo de 1s.

Finalmente la señal es procesada por el bloque “Embedded MATLAB Function” en donde se convierte en un conjunto de pulsos estándar de amplitud +1 como se puede observar en las figuras 2.28 y 2.29. La salida (Out3) entrará a un bloque auxiliar.



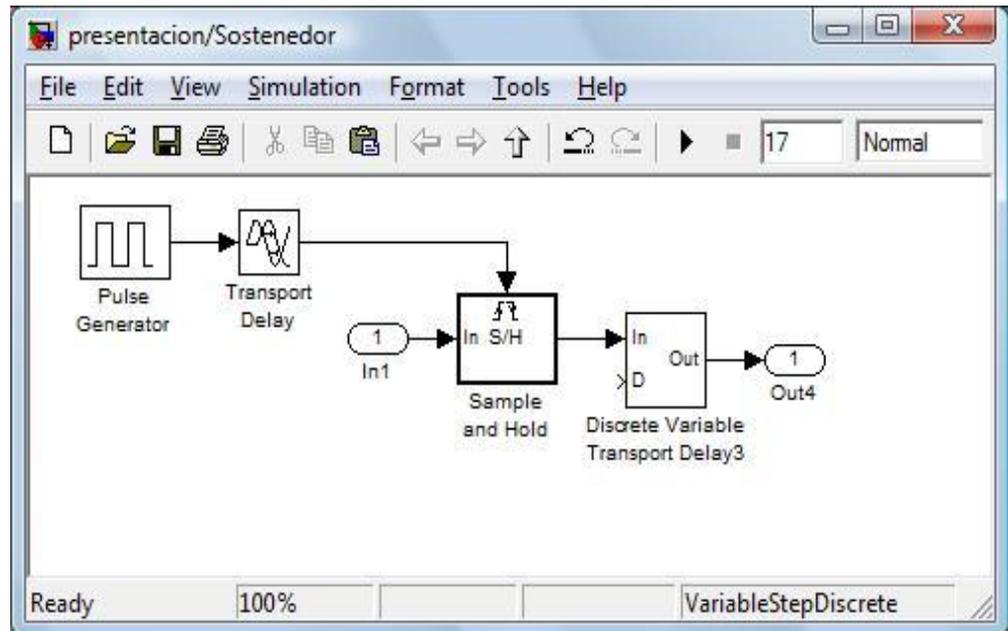
**Figura 2.28** Señal de salida Out3 del Bloque Detector de Ruido a un rango de tiempo de 17s.



**Figura 2.29** Señal de salida Out3 del Bloque Detector de Ruido a un rango de tiempo de  $7 \times 10^{-3}$  s.

### 2.3.5. Bloque Sostenedor

Este bloque actúa como un bloque auxiliar, su función es recibir la señal de pulsos generada en el Bloque Detector de Ruido como entrada, y aumentar el ancho de los mismos para la eficiencia en la conmutación del bloque selector.

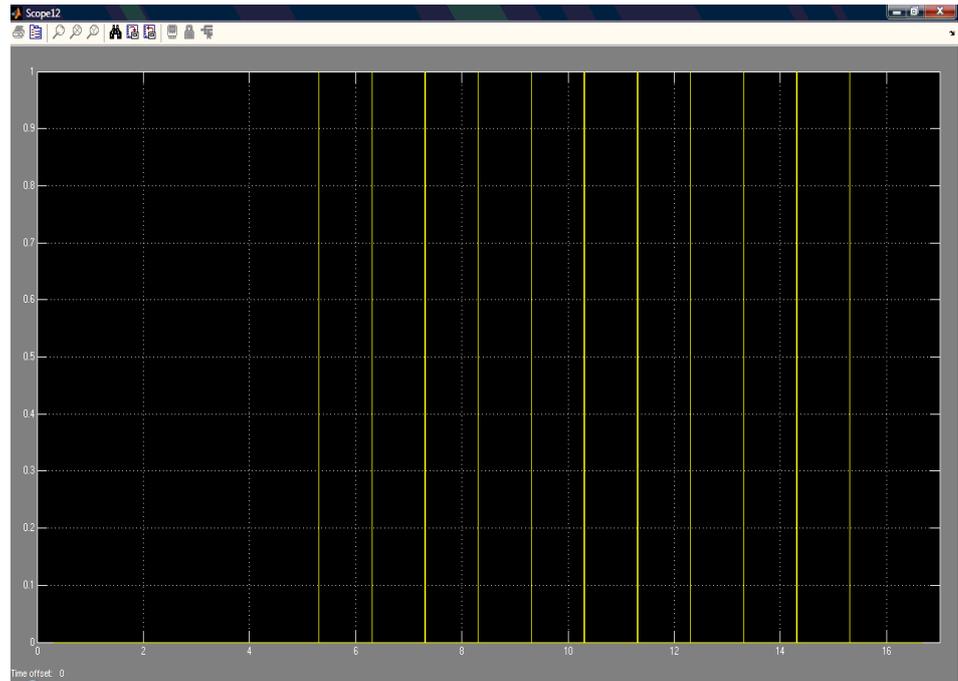


**Figura 2.30** Bloque Sostenedor

La señal de pulsos (In1) entra al bloque “Sample and Hold”, el cual se encarga de sostener la misma; a través de una señal de un trigger o disparo (que es una entrada del bloque).

El trigger o disparo será proporcionado por un bloque “Pulse Generator” es aquí donde se definirá el ancho del pulso necesario para su utilización en el siguiente bloque (Bloque Selector).

Además se tiene un bloque “Transport Delay” con el que se obtendrá un retardo en el generador de pulsos con la finalidad de sincronizarlo con la señal de pulsos originada en el bloque detector (In1) y mantenerla el tiempo de duración necesario (ancho de pulso) para el funcionamiento óptimo del sistema. Ver figura 2.31 y 2.32.



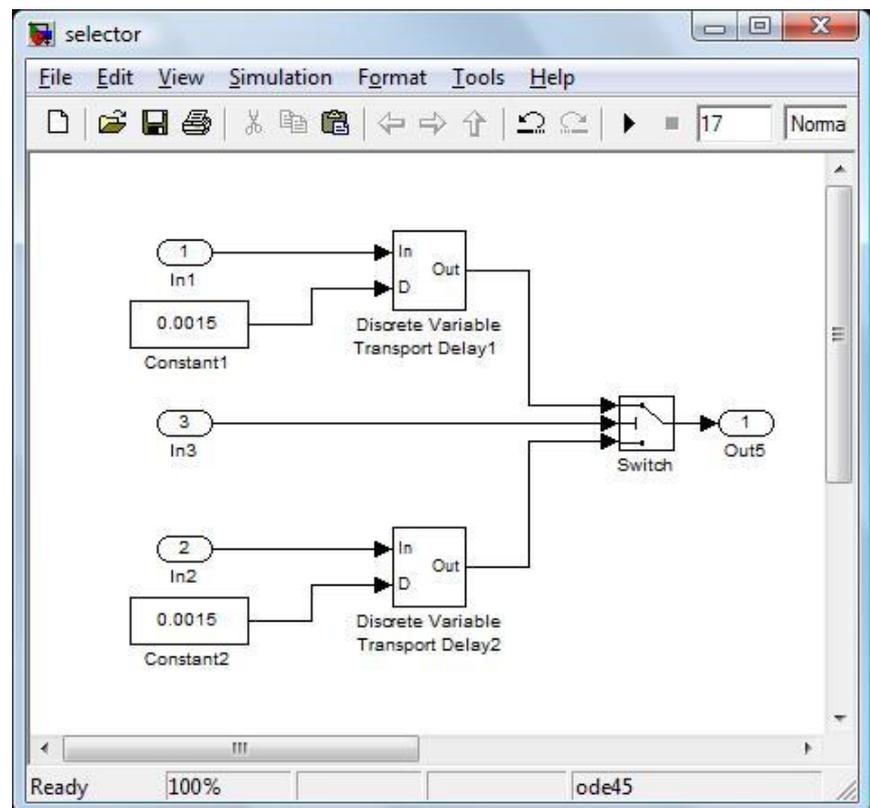
**Figura 2.31** Señal de salida Out4 del Bloque Sostenedor a un rango de tiempo de 17s.



**Figura 2.32** Señal de salida Out4 del Bloque Sostenedor a un rango tiempo de 0.03s.

### 2.3.6. Bloque Selector

Este bloque está conformado por un “Switch” que realiza la conmutación entre la señal de salida Out1 y Out2 del Bloque de Filtrado, a través de una señal de control (Out4) proporcionada por el Bloque Sostenedor. Estas son las entradas In1, In2 e In3 respectivamente. Ver figura 2.33.



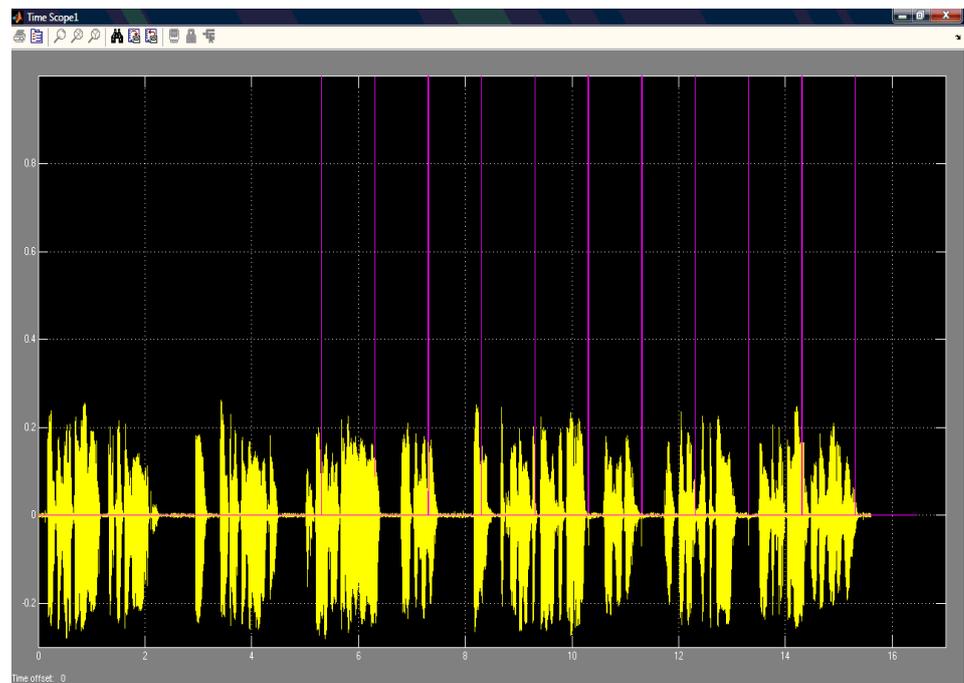
**Figura 2.33** Bloque Selector

La señal In3 durante el proceso de detección de ruido y sostenimiento pasa por varios bloques, lo que produce retardos en la misma antes de llegar al switch, lo cual crea problemas de sincronismo.

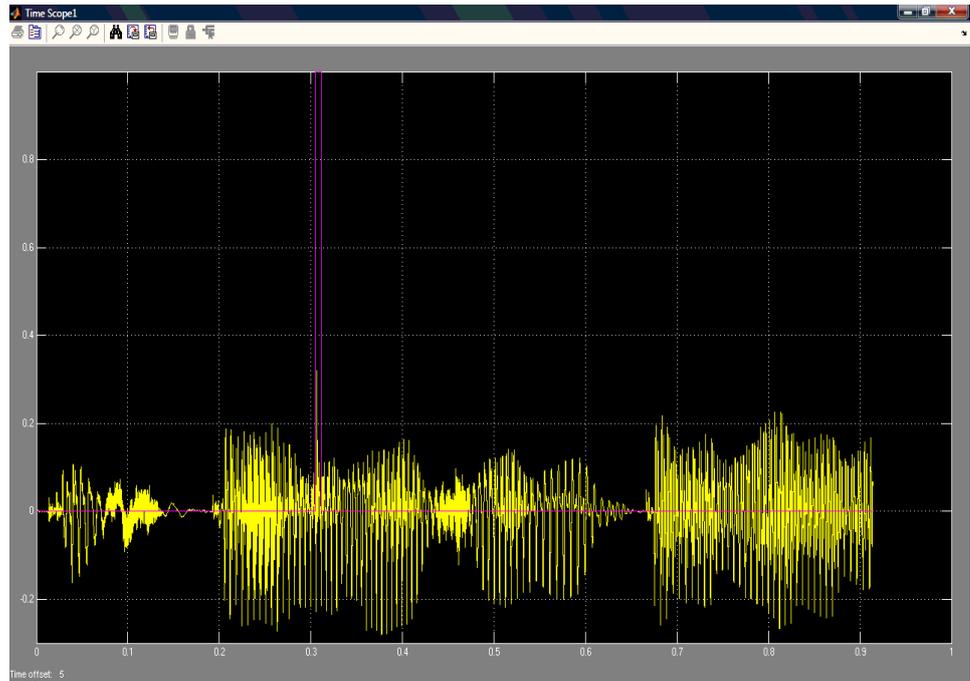
Las señales de entrada In1 e In2 pasan por el bloque “Discrete Variable Transport Delay”, que se encarga de compensar estos retardos al momento de realizar la conmutación.

En este caso el retardo usado es de 0.0015 segundos.

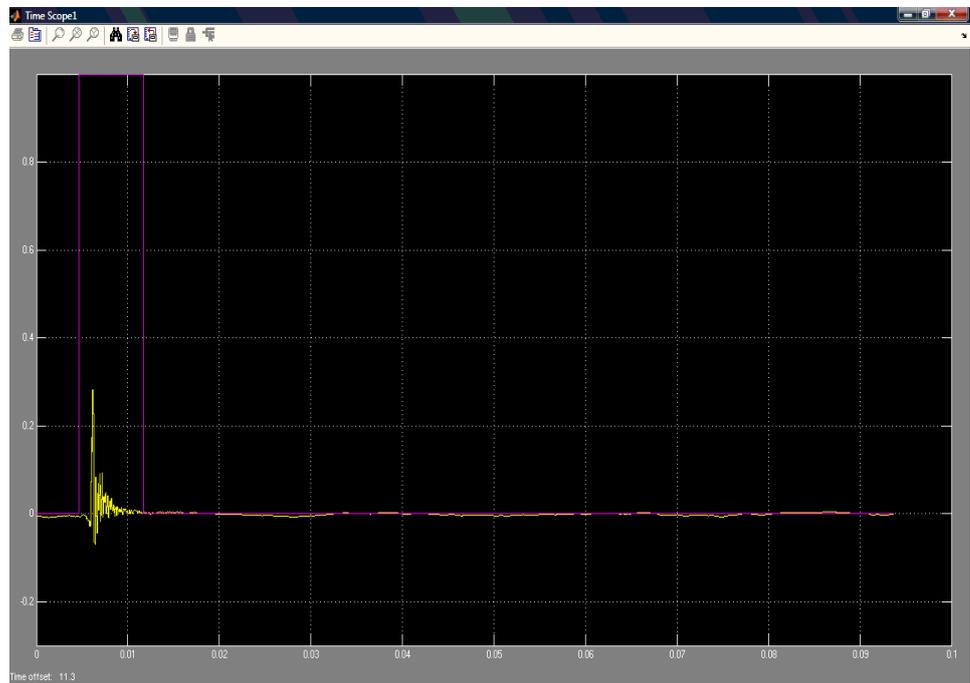
Las figuras 2.34, 2.35 y 2.36 muestran la señal afectada por ruido y la detección del mismo a través de los pulsos de la señal de control; además se puede observar la sincronía para la conmutación.



**Figura 2.34** Detección de señal con ruido a un rango tiempo de 17s.

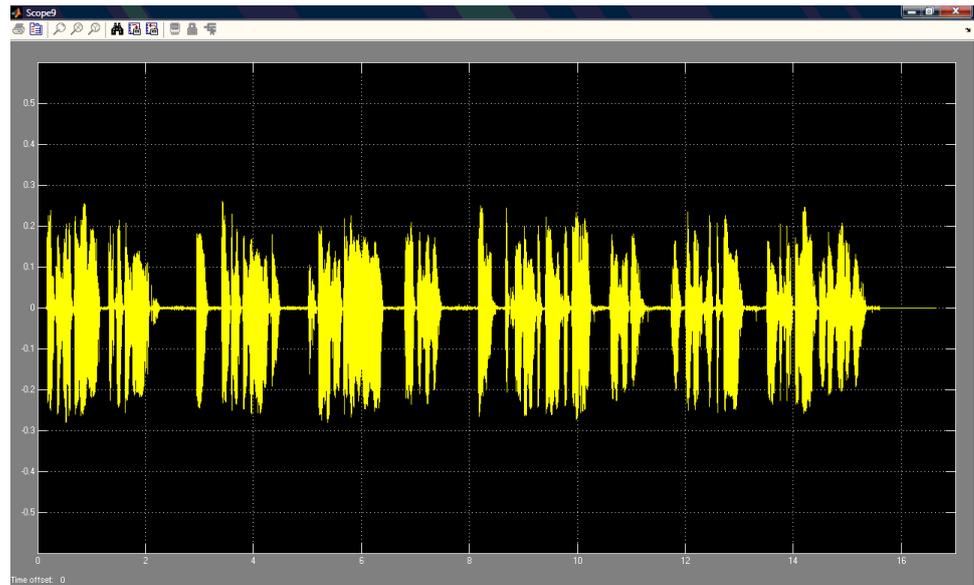


**Figura 2.35** Detección de señal con ruido a un rango tiempo de 1s.

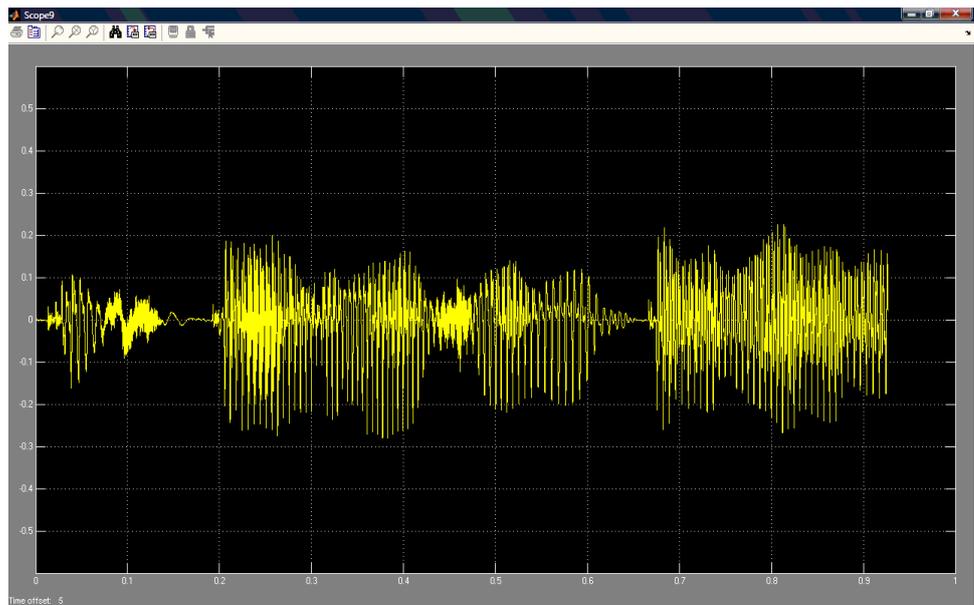


**Figura 2.36** Detección de señal con ruido a un rango tiempo de 0.1s

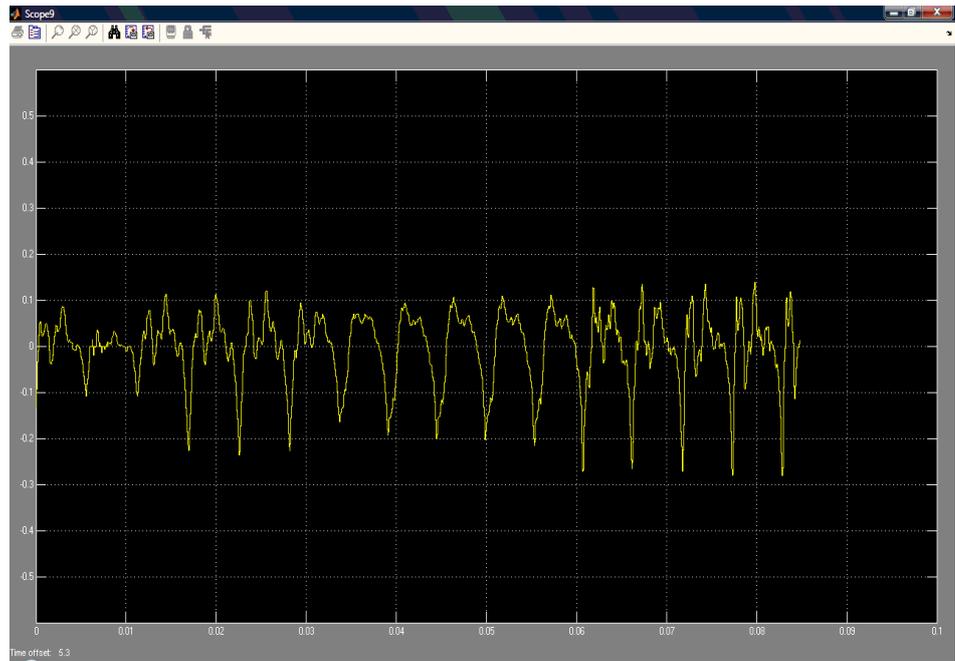
La salida del switch es una señal combinada entre la componente de ruido notoriamente disminuida por el filtro (In1) y la señal no filtrada (In2). Así la salida es del conmutador es una señal mejorada (con respecto al ruido), no suavizada y sin pérdidas de amplitud. Ver figura 2.37, 2.38 y 2.39.



**Figura 2.37** Salida del Bloque Selector a un rango tiempo de 17s.



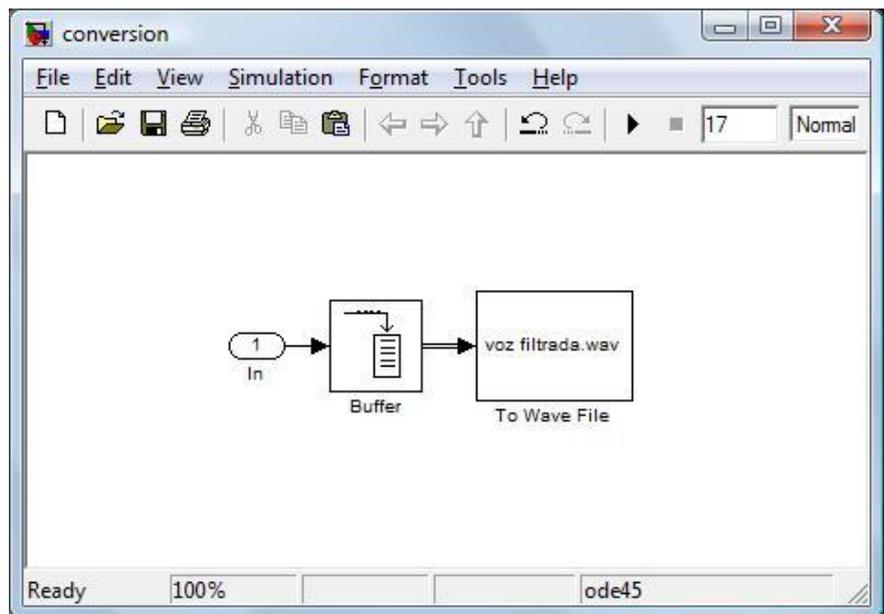
**Figura 2.38** Salida del Bloque Selector a un rango tiempo de 1s.



**Figura 2.39** Salida del Bloque Selector a un rango tiempo de 0.1s.

### 2.3.7. Conversión a un archivo de audio

El bloque encargado de la transformación de la señal de salida del switch a un archivo de audio es “To Wave File”.



**Figura 2.40** Bloques “Buffer” y “To Wave File”.

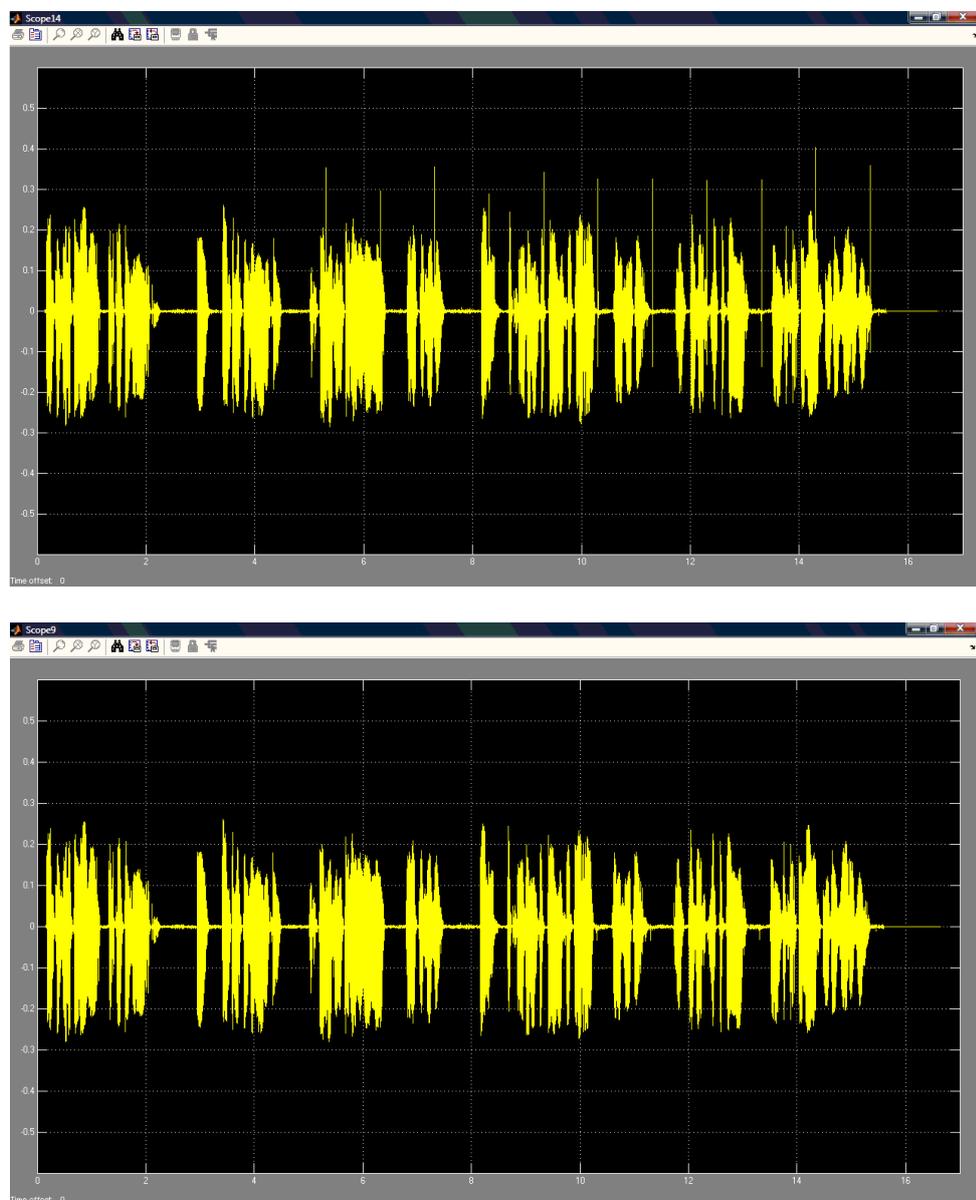
La Salida del Selector ingresa al bloque “Buffer” (In), que cambia la señal de 1x256 basada en muestras, a un vector de 256x1.

Finalmente el vector entra al bloque “To Wave File” que crea el archivo de audio “voz filtrada.wav”; que tiene formato Windows PCM mono con un tamaño de 665 KB, velocidad de transmisión de 320 Kbps a una frecuencia de muestra de datos de 20000Hz.

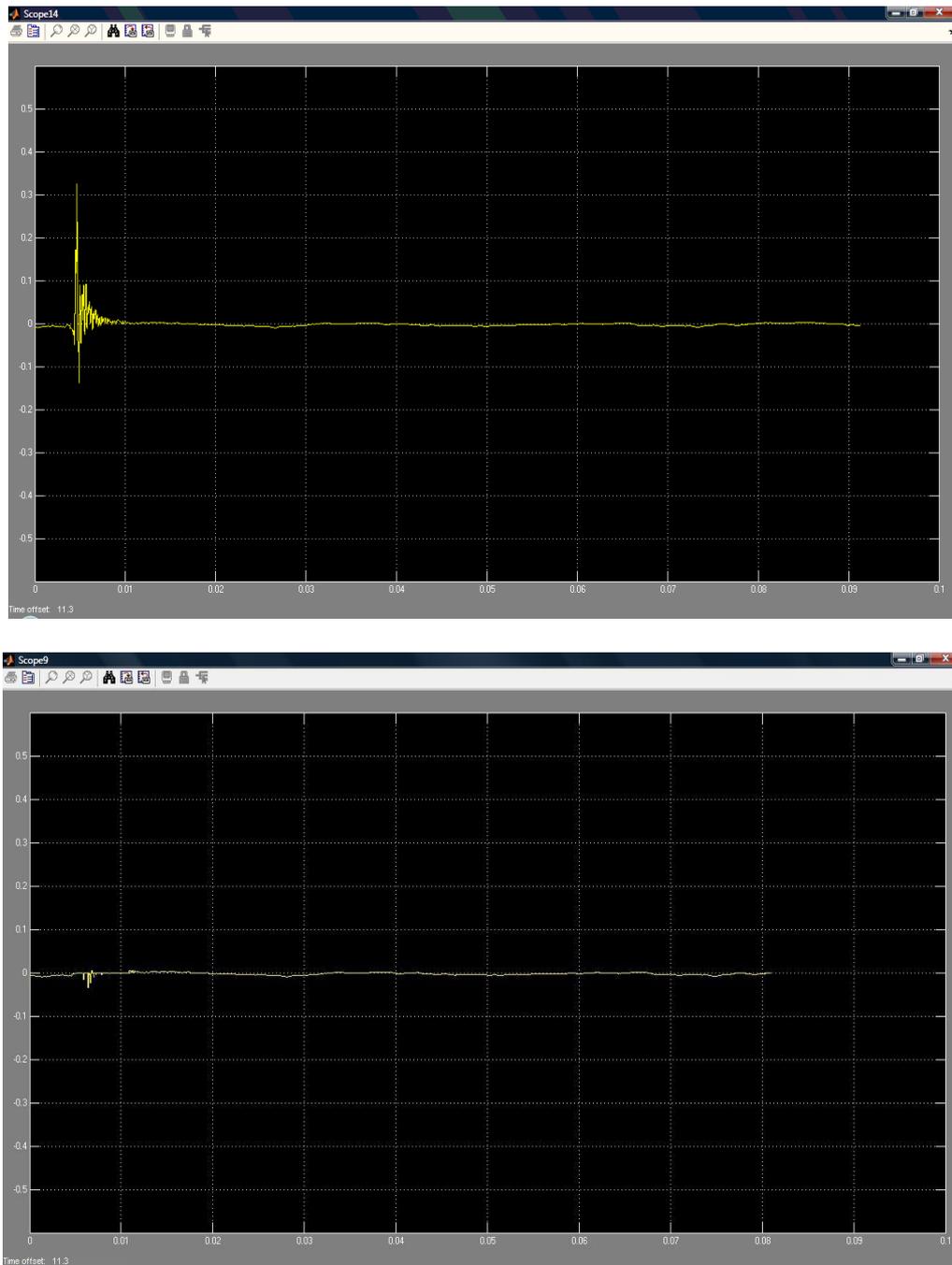
## CAPÍTULO III

### ANÁLISIS DE RESULTADOS

La señal obtenida como resultado en comparación con la señal afectada por ruido de pulsos, muestra la significativa disminución del mismo como podemos observar a continuación.



**Figura 3.1** Comparación entre señal de entrada y salida del sistema en 17s.



**Figura 3.2** Comparación entre señal de entrada y salida del sistema en 0.1s.

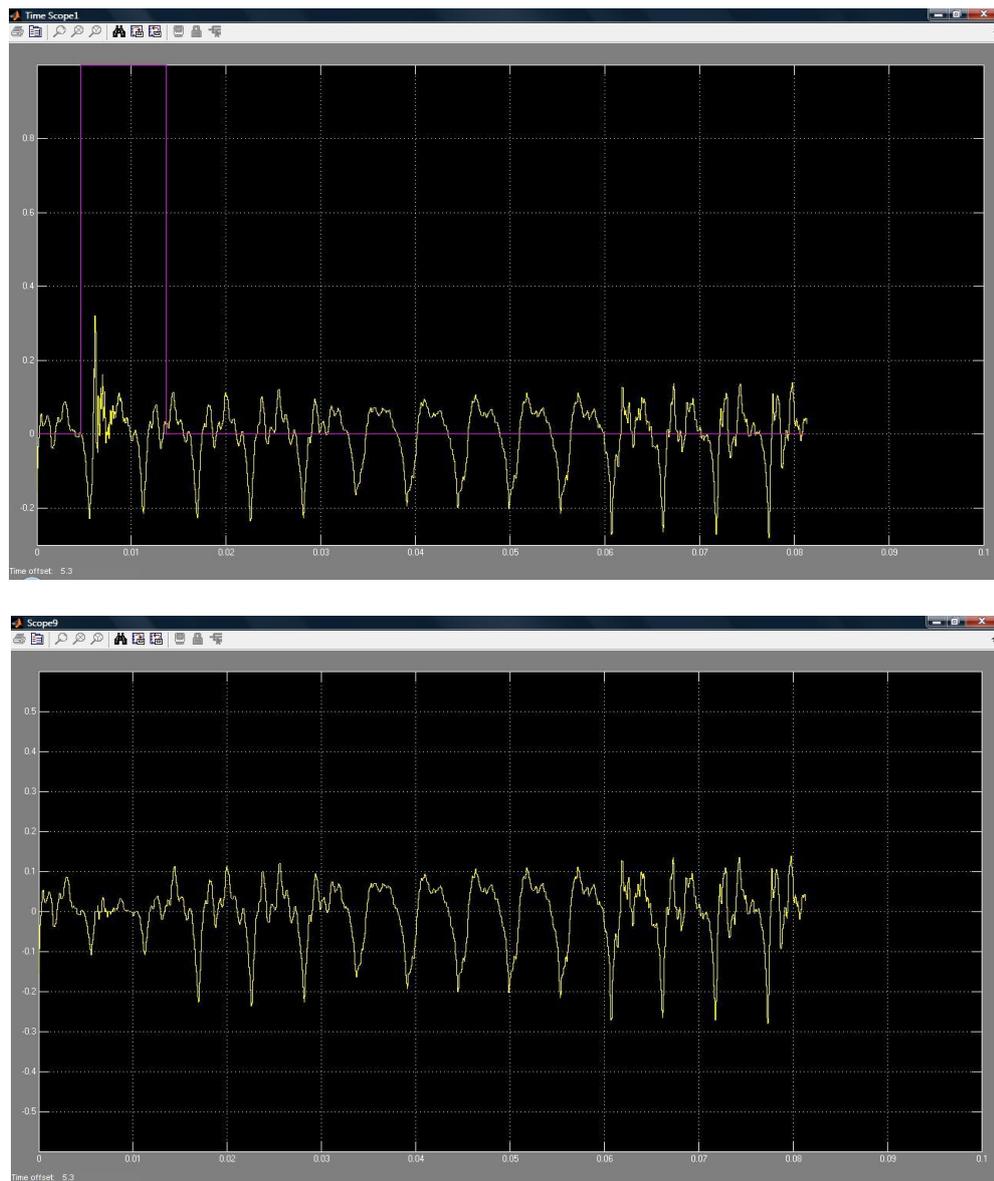
Un factor importante para la eficiencia del sistema es el valor de ancho de pulsos a usar en el Bloque Sostenedor. Para obtener el valor del ancho de pulso eficiente para el sistema, se realizaron varias pruebas, de las cuales se muestran 2 se detallan a continuación.

En la primera prueba el valor del ancho de pulso de 0.003 segundos, en este caso la totalidad del ruido no era filtrado. Ver figura 3.3.



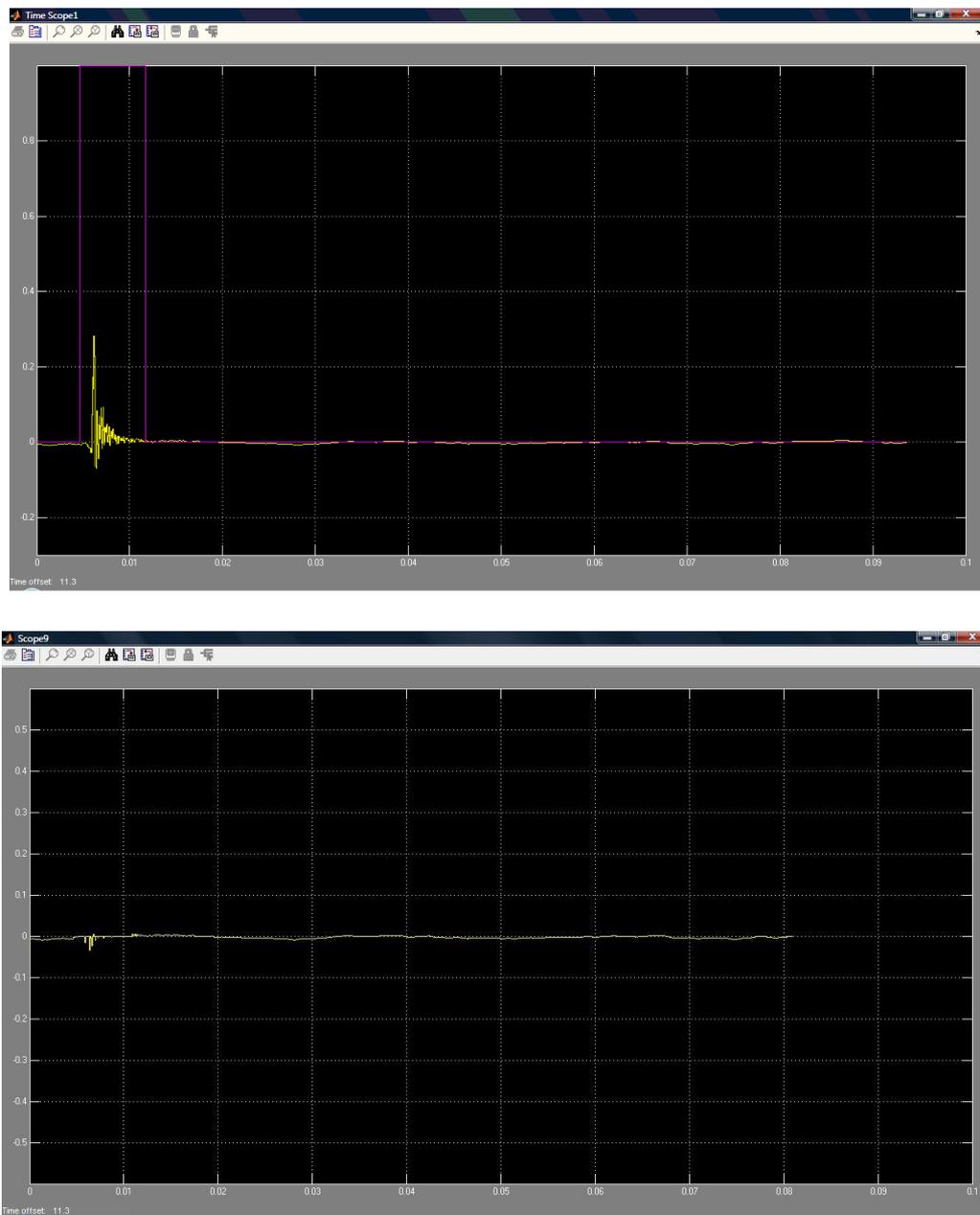
**Figura 3.3** Muestra con un ancho de pulso de 0.003 en 0.1s.

En otra prueba el valor del ancho de pulso de 0.009 segundos, en este caso el pulso es muy ancho, bajando la eficiencia del sistema ya que también abarca componentes de la señal filtrada la cual tiene pérdidas en amplitud. Ver figura 3.4.



**Figura 3.4** Muestra con un ancho de pulso de 0.009 en 0.1s.

Finalmente en la figura se muestra la eficiencia a la salida del sistema (señal disminuida en ruido), usando un ancho de pulsos de 0.00625 segundos; dicho pulso cubre toda la componente del ruido que deseamos eliminar, sin tomar componentes adicionales de la señal filtrada.



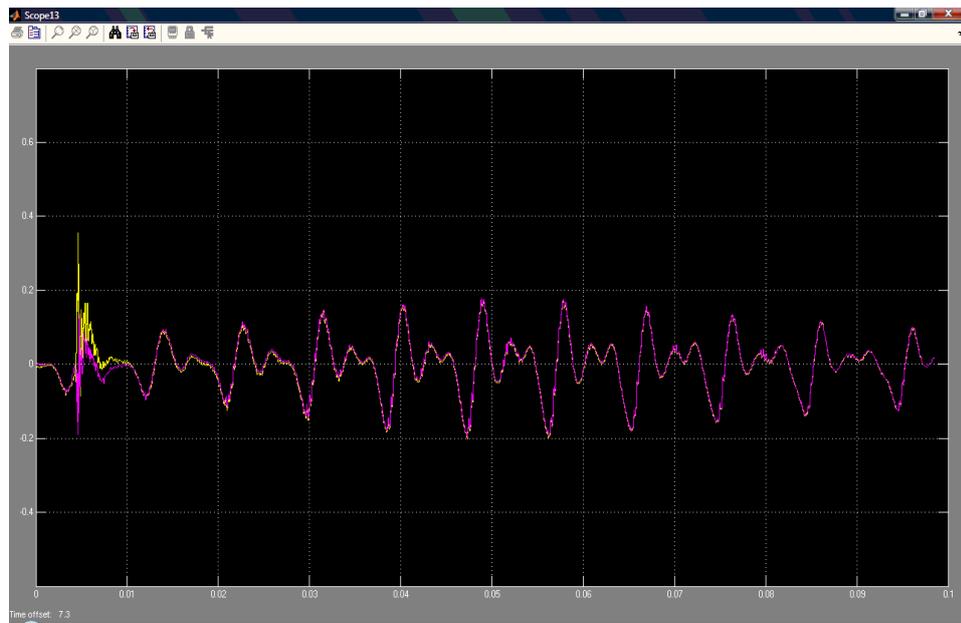
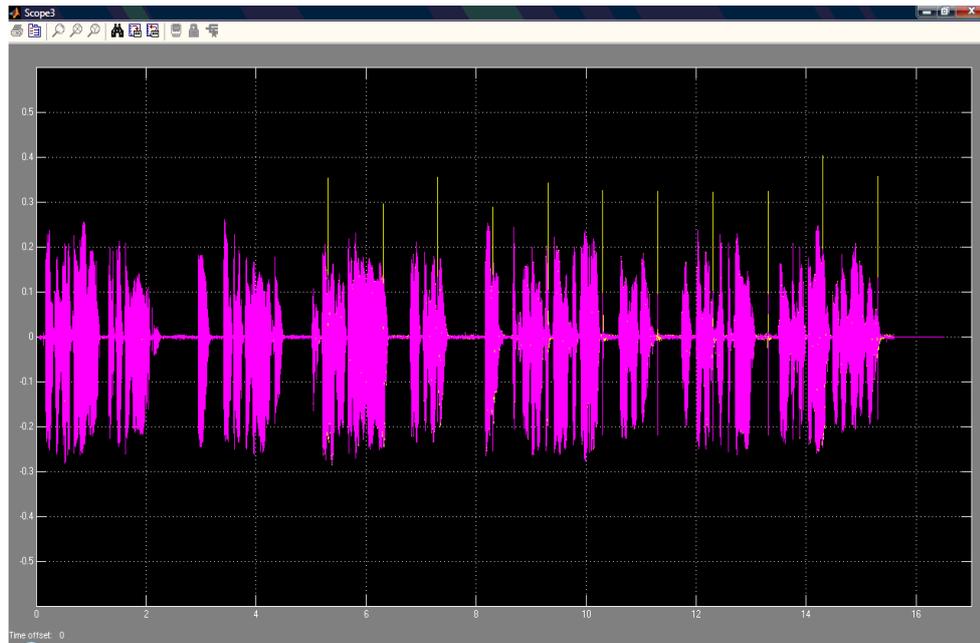
**Figura 3.5** Muestra con un ancho de pulso de 0.00625 en 0.1s.

- Es importante tener en cuenta que para poder seleccionar el valor del *ancho de pulso* correcto para la eficiencia del sistema, en las pruebas que se realizaron además de usar un criterio analítico y visual también se usó un criterio auditivo, es decir, se compararon los archivos “voz con ruido.wav” y “voz filtrada.wav” por cada valor de prueba tomado del

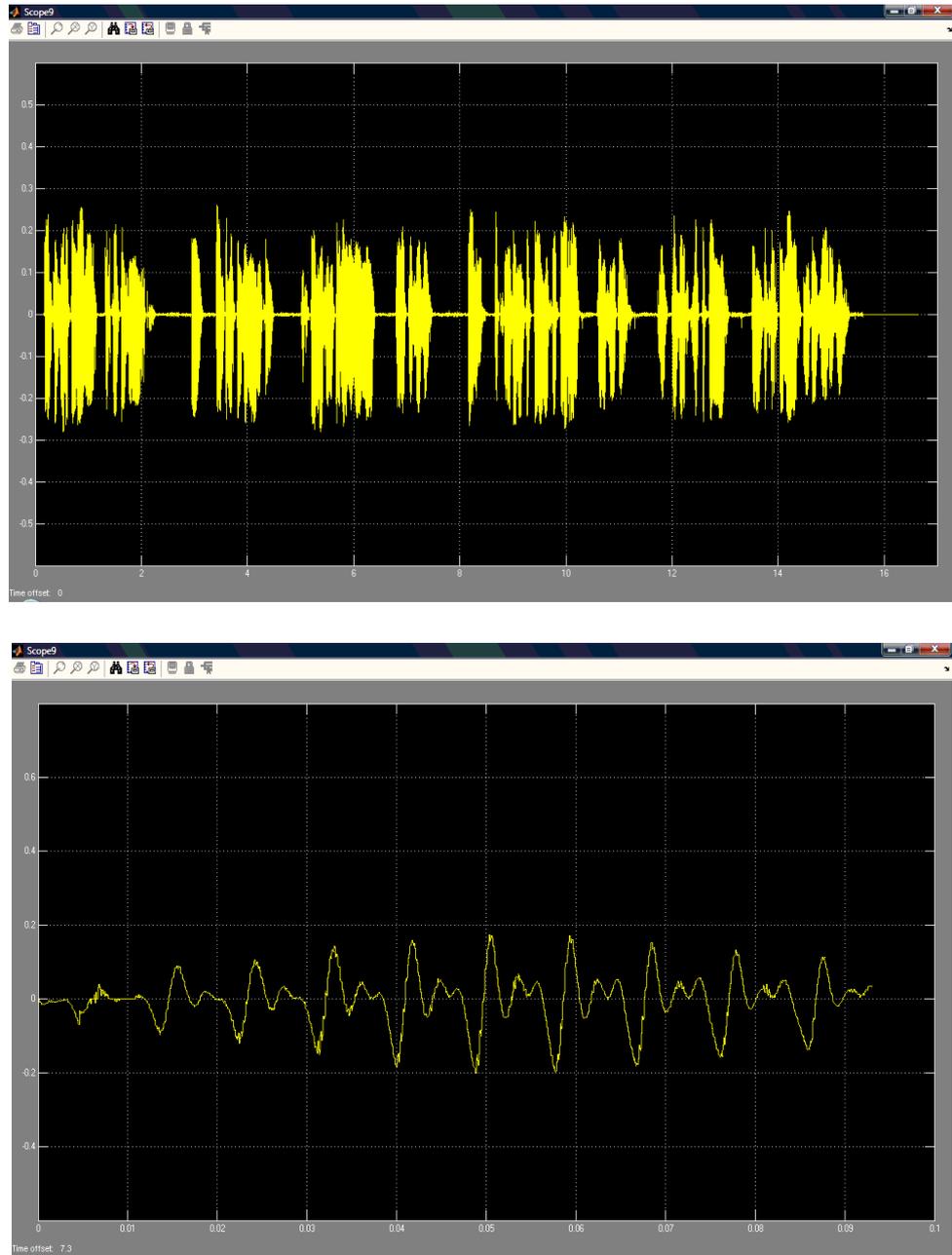
ancho de pulso, hasta determinar el más eficiente en este caso (0.00625 segundos).

Dentro de la implementación del sistema, un aspecto notorio es el cambio de calidad del archivo de audio. El archivo "voz con ruido.wav" es de dos canales con calidad estéreo; en contraste a "voz filtrada.wav" que es de un canal con calidad mono.

Para efectos de esta aplicación, el cambio de calidad en el archivo de audio no es muy relevante, por no ser muy perceptible ante un criterio auditivo. Esta diferencia de canales se puede apreciar en las figuras 3.6 y 3.7.



**Figura 3.6** Señal con ruido, calidad estéreo.



**Figura 3.7** Señal de salida, calidad mono.

## CONCLUSIONES

1. El proyecto presentado en Simulink, herramienta fácil de manejar, tiene un tiempo de procesamiento relativamente bajo y gran eficiencia. Por lo cual es una posible solución para las aplicaciones menores que no podrían tolerar el tiempo de procesamiento de otros métodos más sofisticados
2. Basados en los resultados favorables de las pruebas, descritos en el capítulo “Análisis de Resultados”, se determinó que el filtro de mediana, no solo es funcional para el procesamiento de imágenes, sino que también lo es para aplicaciones de señales de audio afectadas por ruido de pulsos.
3. La transmisión de la información es clara a pesar de que existe una pérdida en la calidad del audio. Esta pérdida es producto del cambio de formato entre el archivo de audio de entrada y de salida del sistema (*estéreo a mono*), sin embargo es poco perceptible ya que se trata de un archivo exclusivamente de voz.
4. Un factor determinante para la graduación de este filtro adaptativo es el ancho de pulso de la señal de conmutación; ya que los resultados experimentales muestran que con un ancho de pulso de 0.00625 segundos se obtiene el máximo rendimiento del sistema.

5. De acuerdo a la percepción auditiva del encuestado, que comparan la señal de audio afectada por ruido con la obtenida a la salida del proceso; el sistema es eficiente. Aproximadamente el 75% de los encuestados consideran a la señal recuperada superior a “Muy buena”; y tan solo un 5% la consideró “Regular”.

## RECOMENDACIONES

1. Dado que los bloques de importación y exportación de datos del programa trabajan con archivos compatibles con el Sistema Operativo de Windows, es recomendable usar este como plataforma de trabajo, para un funcionamiento óptimo y eliminación de conflictos.
2. Para que no se presenten problemas de ruido adicionales causados por distorsión en la grabación, se recomienda usar una entrada de audio de alta fidelidad. La calidad del archivo de audio la garantiza el método de grabación del archivo y la tarjeta de sonido del equipo en el que se trabaja.

## **ANEXOS**

# Anexo A

## Encuesta

**Tabla I** Formato de encuesta

Numero de Encuestados	Calidad del filtrado comparando el audio IN/OUT				
	1 Malo	2 Regular	3 Buena	4 Muy Buena	5 Excelente
1					X
2					X
3					X
4					X
5					X
6				X	
7			X		
8				X	
9				X	
10				X	
11			X		
12					X
13			X		
14				X	
15				X	
16				X	
17				X	
18				X	
19				X	
20			X		
21			X		
22			X		
23		X			
24				X	
25				X	
26					X
27				X	
28			X		
29				X	
30					X
31				X	
32		X			
33				X	
34					X
35				X	
36				X	
37				X	
38			X		
39				X	
40				X	

## Resultados:

**Tabla II** Porcentajes de la encuesta.

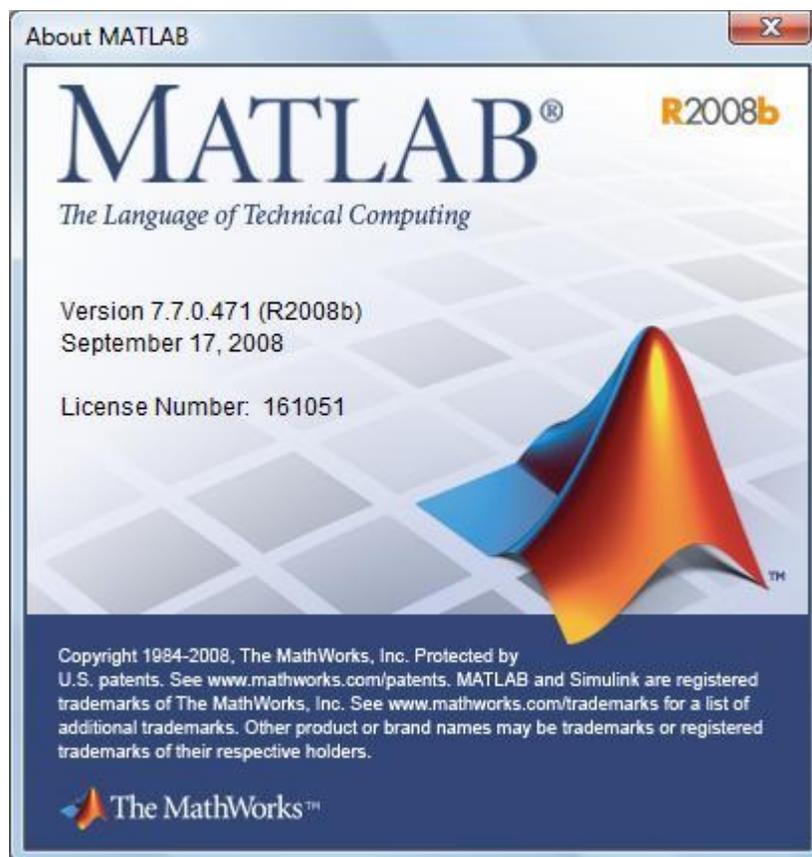
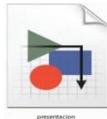
<b>Tabla de Resultados</b>	
<b>Escalas</b>	<b>Porcentajes</b>
Mala	0%
Regular	4%
Buena	20%
Muy Buena	53%
Excelente	23%

# Manual de Usuario

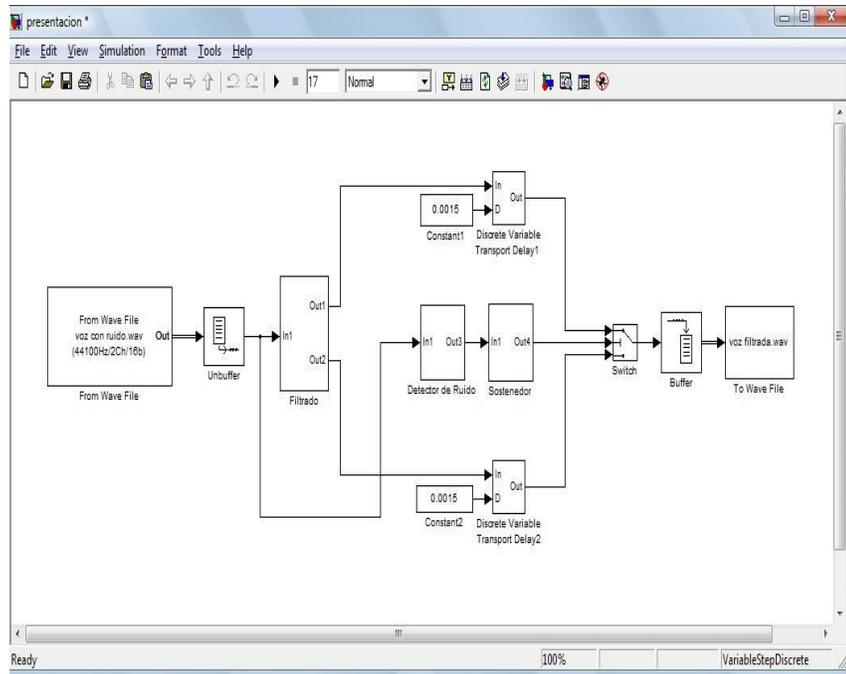
## Restauración de voz deteriorada por ruido de pulsos

Guía para el uso de la interfaz grafica de Simulink desarrollado para la implementación del tema “Restauración de voz deteriorada por ruido de pulsos”.

- Abrir el archivo “presentacion.mdl”



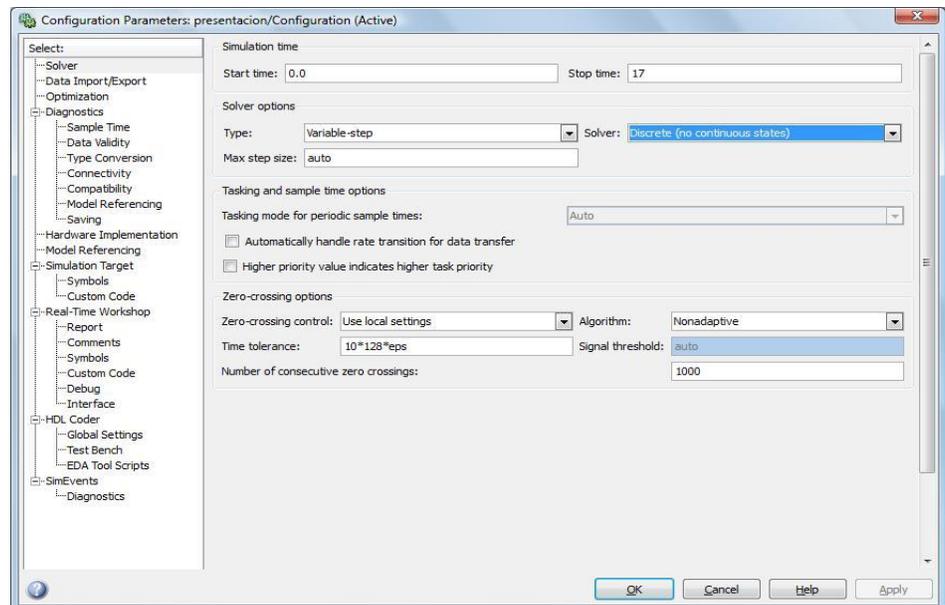
- Aparecerá la siguiente pantalla.



- Se realiza el siguiente ajuste de parámetro:

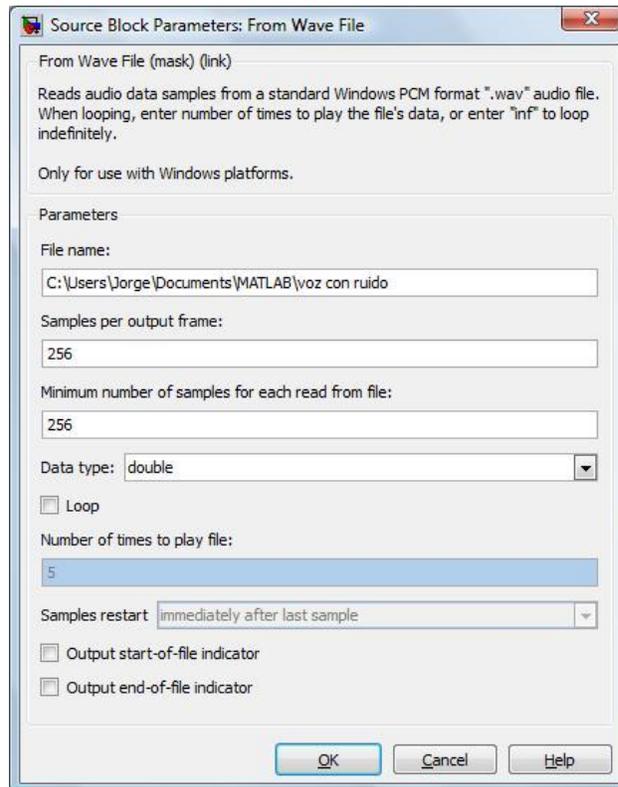
**Simulation;** (de la barra de menú) → **Configuration Parameters**

En el campo **Solver** se selecciona **Discrete (no continuous states)**



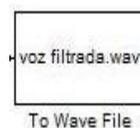
- Se abre la ventana de diálogo del bloque “From Wave File”; en el campo *File Name*, se especifica la ubicación del archivo con el que se va a trabajar.

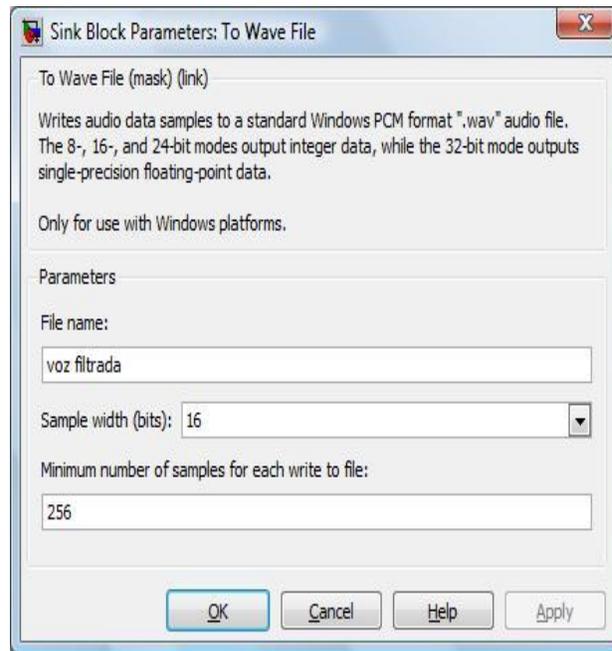
- **From Wave File**



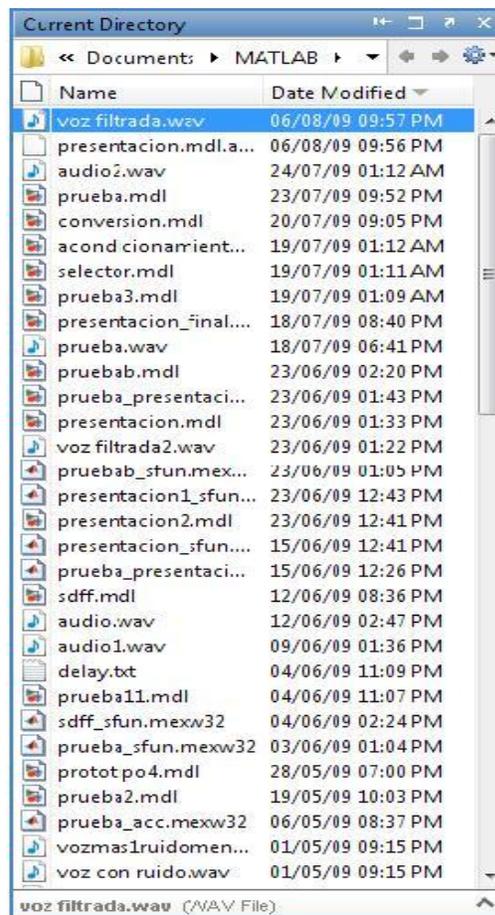
- El nombre del archivo resultante de la simulación, se especifica en el campo *File Name* de la ventana de diálogo del bloque “To Wave File”.

- **To Wave File**





- Este archivo se ubica en "Current Directory" de MATLAB.



- Se especifica el tiempo de simulación deseado (*en este caso 17 segundos*);  
y se presiona “Start Simulation” para dar inicio al proceso.



## REFERENCIAS

1. Rabiner L., Shoter R.; **Digital Processing of Speech Signals**. NJ. Prentice Hall. 1978.
2. Wikipedia, the Free Encyclopedia; **Median Filter**.  
[http://en.wikipedia.org/wiki/Median\\_filter](http://en.wikipedia.org/wiki/Median_filter). 2008.
3. Lasparis T., Lane J.; **Adaptive Scratch Noise Filtering**. IEEE Transactions on Consumer Electronics, vol.39 N.04. 1993.
4. Hodgson R., Bailey D., Naylor M., Ng A., McNeill S.; **Properties, Implementations and Applications of Rank Filters, Image and Vision Computing**. Butterworth and Co. Ltda. vol.3 N.1. 1985.
5. Yli-Harja O., Shmulevich I.; **Efficient Representation and Implementation of Recursive Median Filters and their Cascade Compositions**. 1999.