Sistema de tiempo-real para el procesamiento robusto de señales de voz usando filtrado local adaptativo

Andrés J. Cuevas-Romano, Yuma Sandoval-Ibarra, Victor H. Diaz-Ramirez y Andrés Calvillo-Téllez

Instituto Politécnico Nacional - CITEDI, Avenida del parque 1310, Mesa de Otay, Tijuana B.C. 22510, México
{acuevas,sandoval,vhdiaz,calvillo}@citedi.mx
http://www.citedi.mx/
Paper received on 04/10/12, Accepted on 22/10/12.

Resumen Se presenta un sistema de tiempo-real para el procesamiento robusto de voz, implementado en un arreglo de compuertas lógicas programables (FPGA). El sistema es capaz de estimar una señal de voz limpia a partir de una fuente distorsionada usando un algoritmo basado en el calculo de estadísticas de orden prioritario dentro de una ventana deslizante. El algoritmo incrementa la calidad de voz en términos de métricas objetivas, e introduce únicamente ruido musical imperceptible. El sistema de procesamiento es adaptativo ya que puede variar los parámetros del estimador local usado de acuerdo a los cambios temporales de la señal y de la relación señal a ruido local en cada posición. Se presentan los resultados obtenidos con el sistema propuesto en términos de calidad, inteligibilidad e introducción de ruido artificial, los cuales son discutidos y comparados con los resultados obtenidos con el filtrado de Wiener y el algoritmo de sustracción espectral. Se presenta también, la evaluación del desempeño de tiempo-real del sistema implementado en el FPGA.

Keywords: Mejora de voz, filtrado local adaptativo, estadísticas de orden prioritario, FPGA, sistema en tiempo real

1. Introducción

Debido al increíble avance que han registrado los equipos móviles de comunicación, la demanda por contar con técnicas robustas para la mejora de voz en tiempo-real es hoy en día una necesidad importante. La mejora de voz, consiste en incrementar la calidad de la señal en términos de su inteligibilidad y de la reducción del ruido mediante el uso de métricas de desempeño [2,3]. La mejora de voz es un problema difícil ya que las señales son variantes en el tiempo y además, las funciones de ruido que corrompen a las señales pueden tener un comportamiento estadístico no homogéneo, lo que dificulta la separación efectiva de la voz y el ruido [1]. En la actualidad, existe un gran número de técnicas para la mejora de voz, planteadas desde distintos enfoques teóricos. Algunas de estas técnicas, utilizan un solo canal para separar la señal de voz del ruido. Otras estrategias utilizan un arreglo de sensores (micrófonos) en diferentes

Mireya García-Vázquez, Grigori Sidorov (Eds.).
Advances in Computing Science and Control.
Research in Computing Science 59, 2012, pp. 121- 132

posiciones para resolver el problema. Existe un gran número de trabajos exitosos enfocados a la supresión del ruido en sistemas mono-canal. Cuando podemos asegurar que la función de ruido tiene parámetros estadísticos estacionarios, y su densidad espectral es una constante, el algoritmo de sustracción espectral es la mejor opción [4]. En este enfoque, es necesario estimar el espectro del ruido a partir de la señal capturada de tal manera que en promedio, la relación señal a ruido (SNR) de la señal aumenta [4,5]. Por otra parte, cuando el ruido puede ser modelado como un proceso aleatorio estacionario, la estrategia que es recomendable usar es el filtrado de Wiener [1]. Este enfoque consiste en un sistema lineal que estima la señal de voz a partir de la señal ruidosa minimizando el error cuadrático promedio. Debido a que el filtrado de Wiener y el algoritmo de sustracción espectral se implementan en el dominio de la frecuencia, es muy común que ambos métodos introduzcan artefactos artificiales no deseados como el ruido musical; afectando así la inteligibilidad de la voz. Es importante observar que estas estrategias asumen que el ruido tiene parámetros estadísticos constantes a lo largo del tiempo. Sin embargo, esta suposición no es correcta en la gran mayoría de casos, por ejemplo, ante la presencia de condiciones reales, como ruido de calle o murmullos originados por conversaciones de terceras personas [7]. En este caso es necesario la incorporación de estrategias de estimación robusta para el procesamiento de la señal [8]. Un filtro robusto es usualmente diseñado para solucionar un problema estadístico de estimación con optimización de criterios de desempeño. Al tomar en cuenta las características estadísticas de las señales y del ruido se puede construir un algoritmo de filtrado local adaptativo como función de las estadísticas de orden prioritario de la señal capturada, dentro de una ventana deslizante [7]. Al usar este enfoque, el filtro es capaz de suprimir el ruido conservando los detalles finos de la señal. En procesamiento de voz estas características pueden ser de gran utilidad para aumentar la calidad de la voz sin degradar significativamente la inteligibilidad. En años recientes, los diseños de sistemas basados en FPGA han adquirido una gran popularidad debido a la flexibilidad brindada. Estos dispositivos tienen la posibilidad de reconfigurar el hardware interno conforme a las necesidades del programador y permiten tener un sistema completo dentro del chip del FPGA. Las ventajas de tener un sistema basado en FPGA, son que reduce el tamaño físico del sistema, consume poca energía y debido a que los bloques de procesamiento y los controladores están dentro del FPGA se pueden hacer cambios y mejoras sin necesidad de hacer modificaciones físicas al sistema. En este trabajo se propone un sistema de procesamiento en tiempo-real usando un algoritmo localmente adaptativo basado en estadísticas de orden prioritario para la mejora de voz en un FPGA. La estimación de la señal libre de ruido se realiza aplicando un estimador estadístico sobre las muestras de la señal dentro una ventana deslizante. El algoritmo varia el tamaño y contenido de la ventana así como la función de estimación en relación con las estadísticas locales de la señal ruidosa. Esto significa que el sistema propuesto es capaz de estimar la señal libre de ruido empleando un estimador variante con el tiempo sobre una ventana localmente adaptativa. La ventana adaptativa, es un subconjunto de los elementos de la señal dentro de la ventana deslizante los cuales son cercanos de acuerdo a un criterio específico conveniente respecto a un elemento dado [6]. Consecuentemente, el algoritmo propuesto es capaz de adaptarse a fragmentos no estacionarios de la señal y del ruido y estimar una señal mejorada haciendo uso de un estimador variante en el tiempo. Como resultado,

el método mejora la calidad de voz sin tener que sacrificar la inteligibilidad e introduciendo únicamente ruido musical imperceptible. La organización del documento se describe enseguida. En la sección 2, se presenta una breve introducción a la estimación robusta utilizando estadísticas de orden prioritario. También, se presenta la base teórica del estimador adaptativo utilizado para la mejora de voz, y se presenta el diseño del sistema para el procesamiento de voz en tiempo-real. En la sección 3, se presentan los resultados obtenidos con el sistema propuesto en términos de métricas de desempeño. Finalmente, la sección 4 presenta nuestras conclusiones.

2. Estimación robusta empleando estadísticas de orden prioritario

Consideremos una señal **f** que consiste en la superposición de una señal de voz **s** y la función de ruido **b**, como a continuación:

$$\mathbf{f} = \mathbf{s} + \mathbf{b}. \tag{1}$$

En la Ec. (1), $\mathbf{f}, \mathbf{s}$ y $\mathbf{b}$ son vectores de tamaño $N \times 1$ que representan a las secuencias discretas $f(n)$, $s(n)$ y $b(n)$. Para cada posición-i del segmento **f** podemos crear una ventana deslizante $\mathbf{w}_i$ de tamaño S, como a continuación: $\mathbf{w}_i = [f(n) : |n-i| \leq \frac{(S-1)}{2}]^T$. Estamos asumiendo, que el tamaño S es un numero impar, i es el índice de la posición central de la ventana y T representa la transpuesta del vector. El renglón variacional de $\mathbf{w}_i$ se denota como una secuencia unidimensional $\{\mathbf{V}(r); r = 1,2,\ldots,S\}$, cuyos elementos son ordenados de forma ascendente con respecto a sus valores, es decir, $\mathbf{V}(1) \leq \mathbf{V}(2), \leq \ldots \leq \mathbf{V}(S)$. Los valores $\mathbf{V}(r)$ y $r(V)$ son conocidos como la r-ésima estadística de orden prioritario y el rango del valor V, respectivamente. Ambas cantidades pueden obtenerse del histograma de la señal dentro de $\mathbf{w}_i$ [6,10].

Para crear una ventana adaptativa, es necesario obtener un subconjunto de datos alrededor del elemento central de la ventana deslizante. Existen diversos criterios para construir vecindarios adaptativos, uno de los más utilizados para la reducción de ruido es el vecindario-EV [11]. El vecindario-EV se puede construir a partir de la ventana deslizante como a continuación:

$$\mathbf{v}_i = [\mathbf{v}_i(n) = \mathbf{w}_i(n) : \mathbf{w}_i(i) - \varepsilon_v \leq \mathbf{w}_i(n) \leq \mathbf{w}_i(i) + \varepsilon_v]^T, \tag{2}$$

donde ε_v es un valor constante. En la teoría de estimación robusta existen varios tipos de estimadores de localización de parámetros que se pueden utilizar para estimar el valor del elemento central del vecindario. Nuestra meta es utilizar un estimador robusto para separar la señal de voz de la función de ruido a partir de la ventana adaptativa. El estimador L es uno de los estimadores robustos más populares y se basa en la combinación lineal de estadísticas de orden prioritario. Este estimador es muy popular debido a su simplicidad y gran robustez. El estimador L que se calcula sobre el vecindario adaptativo esta dado por [12]

$$y(i) = \mathbf{a}^T \mathbf{v}_i, \tag{3}$$

donde $\mathbf{v}_i$ es la ventana adaptativa y **a** es un vector de coeficientes de peso, ambos de tamaño $S_A \times 1$. Sea $\mathbf{V}_i$ una matriz diagonal de los elementos del vector $\mathbf{v}_i$. Si calculamos

$\mathbf{R} = \mathbf{V}_i \times \mathbf{V}_i^*$, se obtiene $(y(i))^2 = \mathbf{a}^T\mathbf{R}\mathbf{a}$. Sea x_i un valor de referencia que se asume es muy cercano al valor central de la ventana deslizante de la señal libre de ruido. El error cuadrático entre la estimación y_i y el valor x_i esta dado por $(e(i))^2 = \mathbf{a}^T\mathbf{R}\mathbf{a} - 2\mathbf{a}^T\mathbf{r}$, donde $\mathbf{r} = x(i)\mathbf{v}_i$ es un vector de tamaño $S_A \times 1$. Para obtener un estimador libre de sesgo, debe cumplirse la condición $\mathbf{a}^T\mathbf{u} = 1$, donde $\mathbf{u}$ es un vector unitario de tamaño $S_A \times 1$. Aplicando el método de multiplicadores de Lagrange [12], podemos encontrar el vector de coeficientes como a continuación:

$$\mathbf{a} = \mathbf{R}^{-1}\left[\mathbf{r} + \frac{\mathbf{u}(1 - \mathbf{u}^T\mathbf{R}^{-1}\mathbf{r})}{\mathbf{u}^T\mathbf{R}^{-1}\mathbf{u}}\right]. \tag{4}$$

Finalmente, el vector $\mathbf{r}$ puede calcularse como

$$\mathbf{r} = \mathbf{v}_i\left[\mu_{\mathbf{v}} + S\hat{N}R_i\left(f(i) - \mu_{\mathbf{v}}\right)\right], \tag{5}$$

donde $\mu_{\mathbf{v}}$ es el valor promedio de la ventana adaptativa y $S\hat{N}R_i$ es la relación señal a ruido normalizada "[0,1]" de la señal ruidosa en la i-esima posición de $f(n)$.

2.1. Algoritmo de procesamiento de voz usando estimación robusta

En la Tabla 2.1, se muestra el pseudocódigo del algoritmo propuesto usado para la mejora de voz.

Tabla 1. Psedocódigo y complejidad computacional del algoritmo propuesto.

No.	Instrucción	Tiempo	Complejidad
1.	N = tamaño(**f**)	C_1	$O(1)$
2.	S=tamaño($\mathbf{w}_i$)	C_2	$O(1)$
3.	k = 1	C_3	$O(1)$
4.	for i = 1 to N	C_4	$O(N+1)$
5.	$\mathbf{w}_k = [f(n) : \lvert n-k\rvert \leq (S-1)/2]^T$	C_5	$O(NS)$
6.	**rv** = renglonVaracional($\mathbf{w}_i$)	C_6	$O(NS\ log\ S)$
7.	**v** = conjuntoEV(**rv**)	C_7	$O(NS)$
8.	S_A = tamaño(**v**)	C_8	$O(NS)$
9.	$\mathbf{a}_i$ = calcCoef (**v**)	C_{10}	$O(NS)$
10.	for j = 1 to S_A	C_{11}	$O(N(S_A+1))$
11.	$\mathbf{a}_j^T\mathbf{v}_j$	C_{12}	$O(NS_A)$
12.	if(k < N)	C_{13}	$O(N)$
13.	k = k + 1	C_{14}	$O(N-1)$

De acuerdo al pseudocódigo presentado en la Tabla 2.1, la complejidad computacional del algoritmo se puede expresar por $T(n) = O(NS\ log\ S)$. Observemos que la complejidad computacional del algoritmo adaptativo es polinomial y crece con respecto al tamaño S de la ventana deslizante. Si el tamaño de la ventana es pequeño, el tiempo de procesamiento es corto pero se corre el riesgo de realizar una estimación con información insuficiente en cada posición de la ventana. Si la ventana deslizante es grande,

el tiempo computacional crece considerablemente, sin embargo, la estimación realizada en cada posición de la ventana puede llevarse a cabo usando información suficiente, y así, obtener una estimación precisa. Para tener una buena estimación, se recomienda utilizar el tamaño de la ventana deslizante de al menos dos veces el periodo fundamental de la señal de voz.

2.2. Sistema de tiempo-real usando un FPGA

En esta sección se presenta la implementación realizada en el FPGA del algoritmo adaptativo descrito en la sección 2.1. Primero se realiza una descripción de la configuración del hardware interno del FPGA. Posteriormente, se describe la implementación basada en el procesador embebido Nios II [9]. La tarjeta de desarrollo donde se implemento el sistema de procesamiento de voz, es la tarjeta Altera DE2-115 que cuenta con el chip Cyclone IV EP4CE115. Nuestro objetivó, es configurar un sistema capaz de procesar la señal entrante a velocidad alta; es decir, se requiere que el bloque de procesamiento sea lo suficientemente rápido para que el sistema mantenga un flujo de datos de salida constante, que evite pérdidas por traslape de datos. Para tener el mejor desempeño se usó la configuración Nios II/f (fast) que ópera a mayor velocidad a expensas de utilizar un mayor número de elementos lógicos en el FPGA. Para interconectar los núcleos de propiedad intelectual (IP Cores) se usa la herramienta SOPC Builder. Un núcleo de propiedad intelectual es un modulo de hardware prefabricado que se puede usar en diseños específicos para reducir el tiempo de desarrollo. En la Fig. 1 se observa el diagrama del sistema dentro del FPGA generado con SOPC Builder. En este diseño, se usaron los núcleos de Nios II, JTAG, PLL, controlador (entrada/salida) PIO, Audio, pantalla LCD, y SRAM. Como se observa en la Fig. 1 los núcleos fueron interconectados usando el SIF (system interconnect fabric) que es un sistema de recursos lógicos para interconectar los componentes del software SOPC Builder.

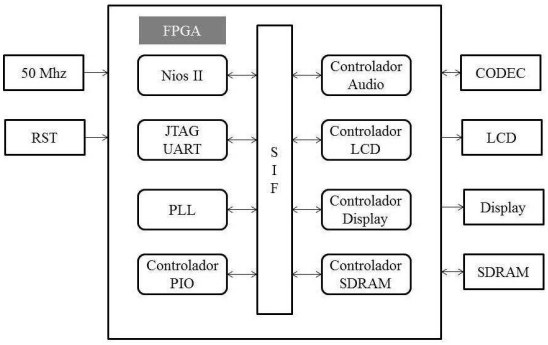

Figura 1. Diagrama a bloques del sistema FPGA usando la herramienta SOPC.

Implementación en Nios II Existen dos tipos de formatos de datos comúnmente usados para el procesamiento digital de señales, punto fijo y punto flotante. Estos formatos

se pueden usar en aplicaciones basadas en FPGAs, sin embargo, en este trabajo se utiliza el formato punto fijo ya que el CODEC de la tarjeta usa datos enteros signados con complemento a dos. El formato punto fijo signado es usado para representar números negativos y positivos, mientras que el formato sin signo solo representa números no negativos. La Tabla 2 muestra el formato numérico usando enteros de 16 y 32 bits. La razón de usar un formato de 16 bits para la implementación es que con un entero de 16 bits se tiene un rango dinámico máximo de 96 dB; que es aproximadamente el rango dinámico del oído humano. En la Tabla 3 se observan algunas conversiones de bits a decibeles.

Tabla 2. Rango dinámico de datos para enteros de 16 y 32 bits.

Formato	Rango
16 bits sin signo	0 a 65,535
16 bits signado	−32,768 a 32,767
32 bits sin signo	0 a 4,294,967,295
32 bits signado	−2,147,483,648 a 2,147,483,647

Tabla 3. Relación entre amplitud y dB.

Amplitud de 16 bits	dB
32,767	0
24,576	−2.4987
16,384	−6.0206
8,192	−12.0412
3,277	−20
1,638	−26.0206
328	−40

La frecuencia de muestreo que se utilizo para procesar las señales de voz es de 8 Khz y los datos son codificados como enteros signados de 16 bits. En la Fig. 2 el CODEC captura la señal de voz entrante a través de la entrada de micrófono (mic) y codifica cada muestra para ser procesada dentro del FPGA. Posteriormente, los datos se dirigen de regreso hacia el CODEC para su reproducción. El algoritmo adaptativo hace su procesamiento en segmentos de datos y se requiere que el sistema haga su procesamiento en alta velocidad, por lo tanto, el bloque de datos no puede ser muy grande ya que tomará una gran cantidad de tiempo para procesar. Adicionalmente, la longitud del segmento no puede ser demasiado pequeña ya que podrían haber interrupciones en el flujo de datos debido a que entrarían mas datos de los que es posible procesar. Es importante notar que el tamaño del bloque de datos es un parámetro ajustable para tener mejor desempeño de latencia o una mejor calidad de procesamiento y depende de la capacidad de procesamiento del sistema. Para la implementación realizada, el tamaño del segmento de datos a procesar es de 200 muestras. Cabe mencionar que 200 datos muestreados a 8KHz/s equivale a un bloque de datos de 25 ms.

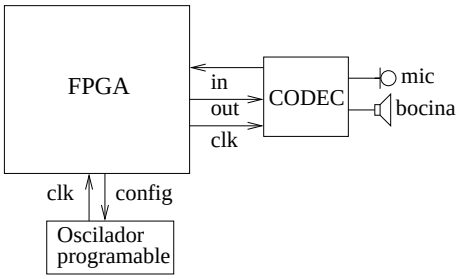

Figura 2. Arquitectura del sistema tiempo-real en el FPGA

3. Evaluación del desempeño del sistema propuesto de tiempo-real

En esta sección se presentan los resultados obtenidos con el sistema propuesto en diferentes implementaciones digitales realizadas. Los resultados obtenidos, son comparados respecto a las siguientes técnicas: algoritmo de sustracción espectral y filtrado de Wiener. La calidad de los resultados esta dada en términos de la métrica de evaluación perceptual de la calidad (PESQ) [15] y la medida de inteligibilidad de tiempo corto objetiva (STOI) [17]. También, se evalúa el índice de artefactos artificiales introducidos por el algoritmo de procesamiento con la métrica SAR (Source Artifacts Ratio), el nivel de distorsión que introducen los algoritmos de procesamiento a la señal con la media SDR (Source to Distortion Ratio), y la capacidad que tienen los algoritmos para suprimir ruido con la métrica SIR (Sourceto Interference Ratio).

3.1. Evaluación del desempeño del algoritmo adaptativo

Para evaluar el desempeño de los algoritmos de procesamiento se usaron los archivos de voz de la base de datos NOIZEUS [14]. Estos archivos, consisten en treinta oraciones producidas por diferentes locutores, capturadas con una frecuencia de muestreo de 8Khz, cuentan con 2^{16} niveles de cuantización y están codificados en formato ".wav". Los archivos de voz se contaminaron con diferentes fuentes de ruido ambiental: ruido de tránsito vehicular, ruido de calle y murmullos con un valor de SNR de 5, 10 y 15 dB. Las señales de voz contaminadas por las fuentes de ruido fueron procesadas con los métodos de substracción espectral, filtrado de Wiener y el algoritmo adaptativo. Cuando la señal de voz esta contaminada con ruido vehicular con un valor de SNR de 15 dB, los parámetros utilizados por el algoritmo propuesto son $S = 65$, $k_1 = 1$ y $k_2 = 0.8$. En la Fig 3 se puede apreciar un ejemplo de una señal procesada con el algoritmo propuesto, en comparación con la señal ruidosa y la señal libre de ruido. También se muestran los espectrogramas de la señal ruidosa y de la señal procesada. La linea negra describe el comportamiento del valor ε_v estimado.

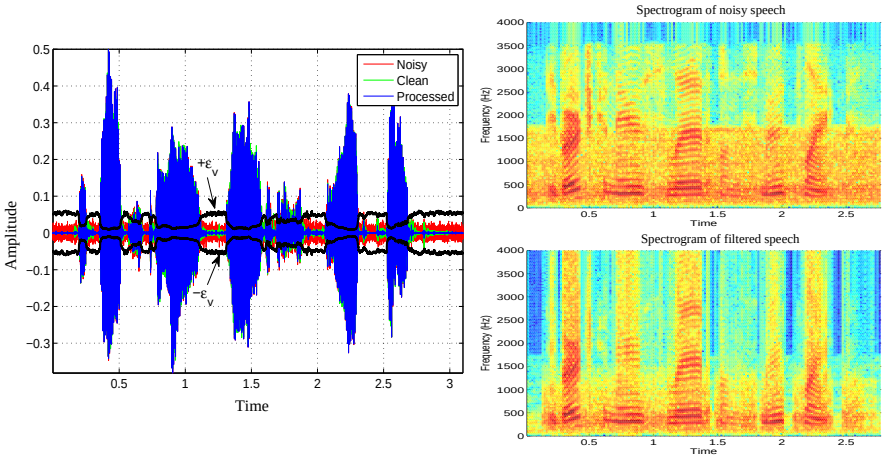

Figura 3. Señal de voz distorsionada con ruido de transito vehicular, su espectrograma y el espectrograma de la señal procesada con el algoritmo propuesto.

Se evaluaron 30 señales de voz con las diferentes funciones de ruido. En las Tablas 4 -12 se muestran los intervalos de confianza con un nivel del 95 % para las diferentes métricas usadas en la evaluación del desempeño de los algoritmos.

Tabla 4. Intervalos de confianza del 95 % cuando la señal de voz esta contaminada con ruido vehicular con 15 dB de SNR.

	Propuesto	**Wiener**	**SpecSub**
SIR	7.72 ± 0.05	19.84 ± 0.31	9.16 ± 0.11
PESQ	2.05 ± 0.01	1.81 ± 0.03	2.08 ± 0.01
STOI	0.75 ± 0.01	0.66 ± 0.01	0.73 ± 0.01
SDR	6.89 ± 0.04	8.51 ± 0.08	7.32 ± 0.07
SAR	15.16 ± 0.06	8.89 ± 0.09	12.47 ± 0.20

Tabla 5. Intervalos de confianza del 95 % cuando la señal de voz esta contaminada con ruido vehicular con 10 dB de SNR.

	Propuesto	**Wiener**	**SpecSub**
SIR	13.70 ± 0.17	18.07 ± 0.78	12.91 ± 0.45
PESQ	2.18 ± 0.04	1.84 ± 0.05	2.21 ± 0.03
STOI	0.80 ± 0.01	0.72 ± 0.01	0.79 ± 0.01
SDR	11.48 ± 0.10	10.28 ± 0.09	10.72 ± 0.11
SAR	15.67 ± 0.10	11.22 ± 0.12	15.37 ± 0.56

Tabla 6. Intervalos de confianza del 95 % cuando la señal de voz esta contaminada con ruido vehicular con 5 dB de SNR.

	Propuesto	**Wiener**	**SpecSub**
SIR	9.33 ± 0.10	19.75 ± 0.50	9.17 ± 0.16
PESQ	2.05 ± 0.01	1.78 ± 0.03	2.05 ± 0.01
STOI	0.75 ± 0.00	0.65 ± 0.01	0.72 ± 0.01
SDR	7.49 ± 0.07	8.50 ± 0.08	7.21 ± 0.07
SAR	12.61 ± 0.07	8.90 ± 0.08	12.17 ± 0.26

Cuando las señales de voz están contaminadas con ruido vehicular el algoritmo que presenta los mejores resultados en términos de calidad (PESQ) es el método de sustracción de espectral, sin embargo, introduce ruido musical muy notorio e introduce un alto nivel de distorsión (ver nivel de SDR). El filtrado de Wiener es el que presenta mejor capacidad para eliminar el ruido (buen nivel de SIR) pero sacrifica inteligibilidad (STOI) y calidad (PESQ), ya que degrada los detalles de la señal. El algoritmo propuesto presenta buenos resultados en términos de calidad (PESQ) preservando la inteligibilidad (STOI), además el porcentaje de distorsión (SDR) es tolerable y no introduce ruido musical perceptible. En las Tablas 7-9 se muestran los intervalos de confianza con un nivel del 95 % de las diferentes métricas de evaluación del desempeño de los algoritmos cuando la señal de voz esta contaminada con ruido de murmullos para un SNR de 15, 10 y 5 dB.

Tabla 7. Intervalos de confianza del 95 % cuando la señal de voz esta contaminada con ruido de murmullo con 15 dB de SNR.

	Propuesto	**Wiener**	**SpecSub**
SIR	18.27 ± 0.14	21.48 ± 0.33	18.20 ± 0.31
PESQ	2.70 ± 0.01	2.23 ± 0.02	2.71 ± 0.02
STOI	0.89 ± 0.00	0.84 ± 0.00	0.90 ± 0.00
SDR	16.51 ± 0.09	12.35 ± 0.08	15.36 ± 0.15
SAR	21.39 ± 0.09	12.98 ± 0.12	19.18 ± 0.70

Tabla 8. Intervalos de confianza del 95 % cuando la señal de voz esta contaminada con ruido de murmullo con 10 dB de SNR.

	Propuesto	**Wiener**	**SpecSub**
SIR	12.90 ± 0.16	18.47 ± 0.61	13.07 ± 0.51
PESQ	2.38 ± 0.02	1.93 ± 0.04	2.39 ± 0.03
STOI	0.82 ± 0.00	0.76 ± 0.01	0.83 ± 0.01
SDR	11.49 ± 0.10	10.32 ± 0.11	10.95 ± 0.13
SAR	17.31 ± 0.14	11.16 ± 0.19	16.09 ± 0.92

Tabla 9. Intervalos de confianza del 95 % cuando la señal de voz esta contaminada con ruido de murmullo con 5 dB de SNR.

	Propuesto	**Wiener**	**SpecSub**
SIR	6.18 ± 0.08	9.68 ± 1.05	6.62 ± 0.41
PESQ	2.17 ± 0.05	1.61 ± 0.07	2.17 ± 0.05
STOI	0.76 ± 0.01	0.64 ± 0.01	0.73 ± 0.01
SDR	5.64 ± 0.08	5.80 ± 0.32	5.50 ± 0.23
SAR	15.90 ± 0.09	8.87 ± 0.20	13.21 ± 0.57

Cuando las señales de voz están contaminadas con ruido de murmullos, el algoritmo propuesto presenta buenos resultados en términos de inteligibilidad (STOI) y calidad (PESQ). Ademas, el porcentaje de distorsión (SDR) es tolerable y el ruido musical introducido por el algoritmo es imperceptible (el mejor valor de SAR). En las Tablas 10-12 se muestran los intervalos de confianza con un nivel del 95 % de las diferentes métricas para evaluar el desempeño de los algoritmos cuando la señal de voz esta contaminada con ruido de calle con un nivel de SNR de 15, 10 y 5 dB.

Tabla 10. Intervalos de confianza del 95 % cuando la señal de voz esta contaminada con ruido de calle con 15 dB de SNR.

	Propuesto	**Wiener**	**SpecSub**
SIR	18.55 ± 0.33	22.62 ± 0.82	18.17 ± 0.57
PESQ	2.51 ± 0.02	2.18 ± 0.04	2.48 ± 0.02
STOI	0.86 ± 0.00	0.79 ± 0.01	0.87 ± 0.00
SDR	16.42 ± 0.17	12.48 ± 0.14	14.93 ± 0.26
SAR	20.66 ± 0.15	13.01 ± 0.23	18.40 ± 0.85

Tabla 11. Intervalos de confianza del 95 % cuando la señal de voz esta contaminada con ruido de calle con 10 dB de SNR.

	Propuesto	**Wiener**	**SpecSub**
SIR	11.96 ± 0.18	17.12 ± 1.27	12.05 ± 0.59
PESQ	2.42 ± 0.04	2.10 ± 0.08	2.43 ± 0.04
STOI	0.81 ± 0.01	0.76 ± 0.02	0.82 ± 0.01
SDR	10.80 ± 0.09	10.45 ± 0.20	10.34 ± 0.11
SAR	17.47 ± 0.28	12.00 ± 0.50	17.32 ± 1.42

Tabla 12. Intervalos de confianza del 95 % cuando la señal de voz esta contaminada con ruido de calle con 5 dB de SNR.

	Propuesto	**Wiener**	**SpecSub**
SIR	6.65 ± 0.11	11.39 ± 1.17	6.68 ± 0.46
PESQ	1.99 ± 0.05	1.48 ± 0.07	1.88 ± 0.05
STOI	0.76 ± 0.01	0.61 ± 0.03	0.70 ± 0.02
SDR	5.98 ± 0.10	6.33 ± 0.20	5.19 ± 0.23
SAR	15.26 ± 0.20	8.93 ± 0.56	13.06 ± 1.61

Podemos observar cuando la señal de voz esta contaminada con ruido de calle, el algoritmo de sustracción espectral presenta buenos resultados en la inteligibilidad (STOI) y calidad (PESQ), pero introduce ruido musical muy notorio y un alto nivel de distorsión (SDR). El filtrado de Wiener tiene mayor capacidad para eliminar el ruido (ver nivel de SIR), sin embargo, distorsiona considerablemente la señal (SDR) e introduce un nivel importante de artefactos artificiales (SAR). El algoritmo adaptativo da buenos resultados en términos de calidad e inteligibilidad, y no introduce ruido musical perceptible. Podemos observar en los resultados obtenidos que el algoritmo propuesto es robusto porque es capaz de adaptarse bien a las características no homogéneas de la señal de entrada así como al comportamiento no estacionario de la funciones de ruido. Además, el sistema propuesto es capaz de eliminar el ruido sin introducir ruido musical perceptible.

3.2. Evaluación del desempeño del sistema en tiempo-real

En esta sección se presenta la evaluación del desempeño del sistema en tiempo-real basado en FPGA. La señal de entrada fue procesada con el algoritmo adaptativo usando los parámetros $S = 35$, $k_1 = 1$ y $k_2 = 0.8$. Para evaluar el desempeño de tiempo-real del sistema se midió latencia del FPGA usando un tono de 600 Hz como señal de entrada. El resultado obtenido es una latencia de 64 ms cuando el filtro esta desactivado, es decir, cuando la señal de entrada al CODEC pasa directamente al bloque de salida a través del procesador NIOS II sin sufrir modificación alguna. Una vez activado el bloque de procesamiento, se introducen 16 ms adicionales para obtener un total de 80 ms de latencia. Estos resultados pueden verse en la Fig.4.

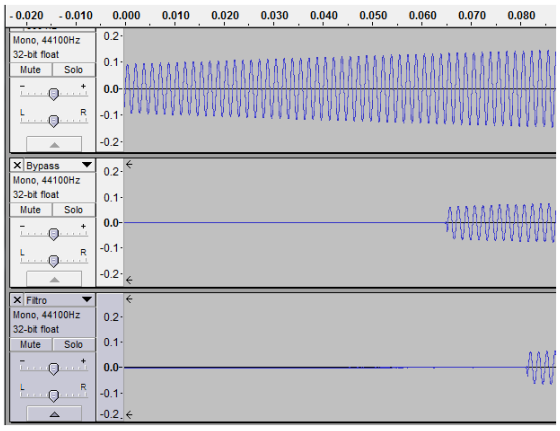

Figura 4. Latencia del sistema: (arriba) tono de entrada de 600 Hz, (centro) salida del sistema sin procesamiento, (abajo) salida del sistema con procesamiento activado.

4. Conclusiones

Se implemento un sistema de tiempo-real basado en un FPGA para la mejora de voz utilizando estimadores robustos de orden prioritario. El sistema es capaz procesar señales de voz a velocidad alta. El algoritmo implementado es capaz de reducir los efectos del ruido sin introducir ruido musical perceptible. En base a las pruebas realizadas, el sistema propuesto es capaz de incrementar la calidad de una señal de voz ruidosa, sin degradar la inteligibilidad e introduciendo bajos niveles de ruido artificial. El sistema implementado en el FPGA, puede ser utilizado para aplicaciones de tiempo-real.

Referencias

1. J Benesty, and S. Makino and J. Chen, *Seepch Enhancement*, Springer Series on Signals and Communitacion Technology, 2005.
2. Philips C. Loizou, *Speech Enhancement: theory and practice*, Taylor & Francis, 2007.
3. Yi Hu and Philips C. Loizou, *Subjective comparision and evaluation of speech enhancement algorithms*, Speech Communication, vol. 49, pp. 588-601, 2007.
4. S. F. Boll, *Suppression of Acoustic Noise in Speech Using Spectral Subtration*, IEEE Transactions Acoustics Speech Singal Process, vol. 27, no. 2, pp. 113-120, 1979.
5. R. J. McAulay, and M. L. Malpass, *Speech Enhacement Using a Soft-Desicion Noise Suppression Filter*, IEEE Transactions Acoustics Speech Singal Process, vol. 28, pp. 137-145, 1980.
6. L. Yaroslavsky and M. Eden, *Fundamentals of digital optics*. Boston:Birkhäuser, 1996.
7. E. Hansler and G. Schmidt, Eds., *Speech and audio processing inadverse environments*, ser. Signals and Communication Technology. Springer, 2008.
8. Jaakko Astola and Pauli Kuosmanen, *Fundamentals of Nolinear Digital Filtering*, Series Edit by Fidker and Phil Mars, 1997.
9. V. L. Richard Boulanger, Ed., *The Audio Programming Book*. The MIT Press, 2011
10. V. Kober and M. Mozerov and J. Alvarez-Borrego and I. A. Ovseyevich, *Rank Image Processing Using Spatially Adaptative Neigborhoods*, Pattern Recognition and Image Analysis. 2001;(3):542-552.
11. Yuma Sandoval Ibarra, Vìctor H. Díaz-Ramírez, and Juan J. Tapia Armenta, Algoritmo de orden localmente-adaptativo para la mejora de señales de voz, Congreso Internacional de Ciencias de Computación CICOMP 10, 2010.
12. Vìctor H. Díaz-Ramírez, and Andres J. Cuevas-Romano, *Speech processing using local adaptive rank-order estimators*, III Encuentro Internacional Académico y de Investigación y VII Encuentro Regional Académico, 2011.
13. H. Hirsch, and D. Pearce, *The Aurora Experimental Framework for the Performance Evaluation of Speech Recognition Systems under Noisy Conditions.* ISCA ITRW ASR2000, Paris, France, September 18-20, 2000.
14. *IEEE Recommended Practice for Speech Quality Measurements* IIEEE Trans. Audio and Electroacoustics, AU-17(3), 225-246, 1969.
15. *Perceptual evaluation of speech quality (PESQ), and objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs.* ITU-T Recommendation P. 862, 2000.
16. *Objective measurement of active speech level* 2000.
17. Cees H.Tall, Richard C. Hendriks, Richard Heusdens, and Jesper Jensen *An Algorithm for Intelligibility Prediction of Time-Frequency Weighted Noise Speech* IEEE Transactions on Audio, Speech and Languaje Processing, vol. 19, 2125-2136, 2011.