

UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE
TELECOMUNICACIÓN



TRABAJO FIN DE GRADO

DISEÑO E IMPLEMENTACIÓN SOBRE FPGA DE UN PEDAL DE EFECTOS DIGITAL

GRADO EN INGENIERÍA DE TECNOLOGÍAS
Y SERVICIOS DE TELECOMUNICACIÓN

JAVIER OTERO MARTÍNEZ
Madrid, Junio 2019

DISEÑO E IMPLEMENTACIÓN SOBRE FPGA DE UN PEDAL DE EFECTOS DIGITAL

Autor: Javier Otero Martínez

Tutor: Pablo Ituero Herrero
Departamento de Ingeniería Electrónica

Me gustaría agradecer a los que me han apoyado todo este tiempo: mis padres, mi familia, Aida e Impostor mientras trabajaba en el proyecto. También me gustaría agradecer especialmente a los profesores que me han ayudado desinteresadamente cuando he acudido a ellos con alguna pregunta: Alfredo Sanz, Jose Parera, Fernando Gonzalez y por supuesto a mi tutor, Pablo, por toda su paciencia estos meses.

Resumen

El presente proyecto se enmarca en el ámbito de la producción musical en tiempo real. En concreto plantea el diseño, desarrollo e implementación de un pedal de efectos digital. Estos dispositivos permiten añadir en tiempo real efectos y modulaciones a ciertos instrumentos musicales, incluyendo la voz humana. Un octavador es un dispositivo electrónico que recibe un sonido y mediante un procesamiento de señal, entrega a su salida ese mismo sonido una octava por debajo del original y va a ser el efecto principal que se va a implementar. El proyecto plantea el problema desde cero, utilizando la literatura existente para abordar tanto el diseño como la implementación.

Abstract

This is the Abstract

Índice general

1. Introducción y generalidades	1
1.1. Objetivos	2
1.2. Metodología	2
2. Entrada: etapa analógica	3
2.1. Selección de micrófono	3
2.2. Preamp	5
2.3. Alimentación	6
3. El algoritmo	8
3.1. Aproximaciones al problema	8
3.2. Vocoder	8
3.2.1. Vocoder en la música	9
3.3. Transformación a frecuencia: STFT	10
3.3.1. Transformada de Fourier: FFT e iFFT	10
3.3.2. Solapamiento y enventanado	11
3.4. Operaciones sobre la fase	14
4. Implementación	15
4.1. Gestión entrada-salida	15
4.1.1. Pmod i2s2	15
4.1.2. Consideraciones de implementación	15
4.2. Controlador de datos	15
4.2.1. Bancos de memorias	15
4.2.2. Core FFT	15
4.3. Controlador global	15
4.3.1. Displays	16
5. Pruebas y depuración	17

Capítulo 1

Introducción y generalidades

Un pedal de efectos es un dispositivo que se conecta entre un instrumento (normalmente electrófono) y su amplificador, modificando la señal de entrada y sus características fundamentales como pueden ser timbre, tono y volumen. En mi caso particular he optado por un pedal para saxofón, puesto que es el instrumento que yo toco, aunque resulta igual de válido para cualquier otro instrumento cuyo sonido se pueda captar con un micrófono convencional. Generalmente, no es habitual el uso de pedales de efectos en instrumentos de viento debido a que se suele buscar un sonido limpio y fiel. No obstante, en multitud de producciones se pueden apreciar efectos añadidos en postproducción ya sea digital o analógicamente. Algunos ejemplos son el *reverb*, *chorus* u *octavador*. Es este último el efecto que se implementará en este proyecto, con la salvedad de que funcionará en tiempo real.

Se ha decidido el combinar este efecto con la posibilidad de implementarlo en formato de pedal el cual se ha hecho muy popular desde su aparición, debido a que los intérpretes pueden activarlo con el pie pudiendo mantener las manos en el instrumento. En el caso del saxofón, esto no suele ser un problema, ya que los músicos no precisan de estar tocando de forma continuada, si no que suele haber un tiempo para descansar suficientemente largo como para operar el dispositivo que se este utilizando. A pesar de ello, resulta conveniente este formato por analogía con otros instrumentos.

Este proyecto abarca todo el proceso desde la idea inicial de diseño hasta el montaje del prototipo final, por tanto, las especificaciones de funcionamiento que se han utilizado pretenden facilitar un uso profesional del prototipo, de forma que sea compatible con los estándares establecidos en el contexto musical e ingenieril al mismo tiempo.

En primer lugar se utiliza un transductor para adquirir la señal, en este caso, un micrófono convencional. Una vez que el estímulo externo es transformado en pulsos eléctricos, atravesará un etapa de entrada analógica que pre-amplifica la señal y la adecua a la entrada de la FPGA.

Para el prototipo se ha utilizado la placa proporcionada por el departamento: Nexys A7. Esta placa monta una FPGA *Xilinx XC7A100T-1CSG324C* junto con varios switches, botones, leds y displays de 7 segmentos, que harán más fácil el manejo del prototipo. Esta placa tiene un micrófono integrado, pero es de tan baja calidad que se opta por diseñar la etapa de entrada, analógica completamente, y conectarla con uno de los puertos del *Pmod i2s2*, también de Digilent y proporcionado por el departamento. Este módulo contiene

ADC, DAC y los conectores de mini-jack estándar en formato de audio, que servirán para gestionar la señal de entrada y la de salida.

Se implementará un algoritmo que se encargará de llevar a cabo la octavación de la señal de entrada y de proporcionarla en la salida. Este algoritmo utiliza una aproximación de *Phase Vocoder* muy común en el tratamiento de señales de audio realizando una transformación al dominio de la frecuencia mediante FFT. Durante todo el proceso se priorizará el criterio de la baja latencia, dado que si no, resulta imposible la interpretación para el músico. El algoritmo se describirá en profundidad en su capítulo correspondiente.

1.1. Objetivos

Para la realización de este proyecto de forma satisfactoria se ha establecido la consecución de una serie de objetivos que son los siguientes:

- Diseño de un sistema completo, a partir de una problemática definida previamente mediante el estudio del instrumento y las señales.
- Implementación sobre la FPGA proporcionada.
- Construcción de un prototipo para estudiar la problemática desde la práctica.
- Afianzar, debido a todo esto, conocimientos adquiridos durante el Grado en diversos ámbitos como el procesamiento de señal y en concreto en la especialidad de Sistemas Electrónicos como la programación hardware, el montaje de un circuito analógico y la correcta comunicación entre todos los módulos.

1.2. Metodología

- En primer lugar se seleccionará el efecto que se quieren implementar, dando prioridad al octavador.
- Como segundo paso, se estudiará la literatura existente y se probarán distintas soluciones algorítmicas empleando MATLAB.
- Estudio y elección de la interfaces de entrada y salida. Selección de micrófono, ADC y DAC.
- Desarrollo y verificación en VHDL empleando la herramienta VIVADO de Xilinx.
- Montaje de un prototipo empleando la placa Nexys A7 de Digilent, el micrófono seleccionado anteriormente y el resto de dispositivos que fueran necesarios.
- Test y depuración.

Capítulo 2

Entrada: etapa analógica

El pedal de efectos tendrá una estructura clara: la parte más importante es la etapa de procesado, en la que se transforma la señal entrante en la que se va a proporcionar en la salida. La etapa de salida adquiere gran relevancia en sistemas que se diseñan para colocarse en cascada, que no es el caso, no obstante, en este capítulo se habla sobre la etapa de entrada. Esta parte del sistema se encargará de captar la señal y prepararla para su procesado posterior por la FPGA.

2.1. Selección de micrófono

Como normalmente se utilizan pedales de efectos en instrumentos electrófonos, la señal de salida del instrumento ya viaja por un cable de camino a la amplificador por lo que el pedal actúa como un intermediario entre ambos. Sin embargo, en instrumentos de viento, es necesario utilizar algún transductor que sea capaz de convertir la señal acústica consistente en ondas de presión en una serie de impulsos eléctricos que puedan ser editados posteriormente.

La solución más sencilla consiste en utilizar el micrófono que viene integrado con la placa Nexys A7: modelo *ADMP421* de Analog Devices [Nex]. No obstante, la utilización de este micrófono plantea dos problemas.

En primer lugar, es inmediato pensar que incluso en el caso de un prototipo, si se plantea usar como pedal, no resulta nada recomendable colocar el micrófono encargado de recoger todo el sonido en el suelo. Además de estar lejos de la fuente sonora, capturaría el sonido resultante de la manipulación de los controles suponiendo una bajada en la calidad del sonido que pudiese proporcionar el dispositivo. Por tanto, es mejor utilizar un micrófono no integrado al propio diseño.

En segundo lugar, pero no menos importante, conviene tener en cuenta la respuesta de sensibilidad del micrófono será omnidireccional (ver figura 2.1), es decir, captará todos los sonidos sin importar la dirección de dónde vengán. Este tipo de transductores se usan principalmente en radio y televisión, donde puede haber varias personas hablando en el mismo micrófono o para la grabación de orquestas o agrupaciones en localizaciones cerradas determinadas. Estos micrófonos son capaces de captar tanto el sonido proveniente de la fuente directamente como los ecos y reflexiones característicos del espacio, dando la

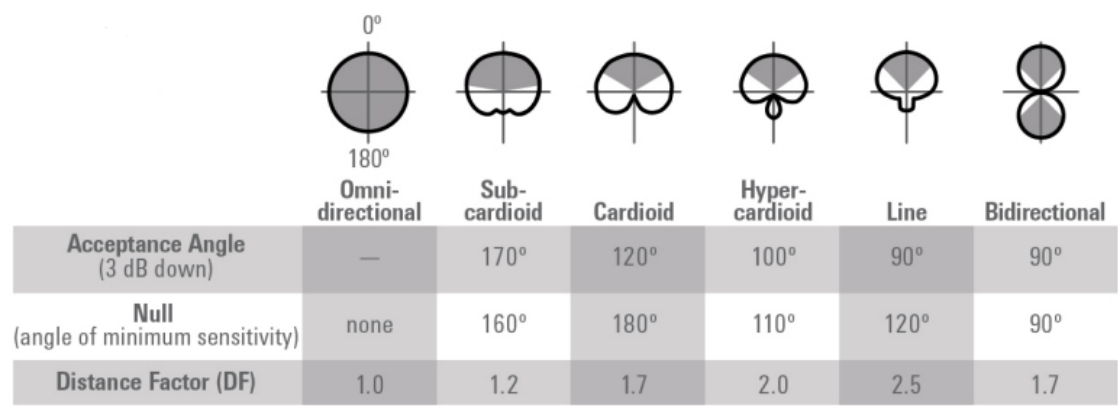


Figura 2.1: Tipos de micrófonos según su diagrama polar [Aud]

sensación de amplitud al oyente que la que produciría su escucha en esa misma localización. Pero mientras que en estos casos se trata de un efecto deseado, resulta poco agradable captar estas reflexiones en un ambiente no preparado para ello y encima cercano al suelo, siendo más conveniente utilizar micrófonos de tipo cardioide.

Estos micrófonos son muy utilizados con instrumentos de viento, ya que el sonido suele provenir de un punto concreto. Es preciso matizar que en el caso del saxofón, contra la creencia popular, el sonido no sale siempre por la campana del instrumento. El sonido sale, por el contrario, por la llave abierta más próxima a la embocadura, que suele ser en torno al centro del instrumento. En consecuencia, no resulta conveniente acercar el micrófono mucho a la campana descuidando otras llaves del instrumento. La figura 2.2 muestra un saxofón alto.

Además de tener en cuenta el diagrama polar para la elección de micrófono, resulta imprescindible conocer los posibles tipos en cuanto a fabricación y funcionamiento. En este sentido tenemos los dos tipos más comunes: los micrófonos de condensador y los dinámicos.

Los micrófonos de condensador reciben su nombre del condensador que poseen en el interior de su cápsula. El diafragma es la pieza encargada de vibrar cuando las ondas de presión del sonido lo atraviesan pero, a su vez, se une esta pieza con una de las placas del condensador. El resultado es que con cada movimiento del diafragma, varía la distancia entre las placas y en consecuencia, se modifica la capacidad de manera inversamente proporcional. Estos cambios modifican una señal eléctrica en la que quedan registradas las ondas de sonido recibidas. Estos micrófonos poseen una buena sensibilidad pero necesitan de alimentación *phantom* de entre 24 y 48 V, que se realiza desde la mesa de mezclas por el mismo cable de señal.

Los micrófonos dinámicos por el contrario, no requieren de alimentación y además poseen buena robustez y son más baratos que los anteriores. Estos funcionan gracias a una bobina unida al diafragma que se mueve conforme a las ondas de presión recibidas del sonido dentro de un campo electromagnético. Por la acción de la inducción electromagnética, este movimiento genera una corriente que atravesará la bobina proporcionalmente al estímulo entrante. La principal desventaja de estos micrófonos es que su respuesta no es del todo lineal con la frecuencia, produciendo mayor o menor ganancia en función del rango del espectro en el que se encuentre el sonido de entrada. Para arreglar esto se suele

usar ecualización posterior o diferentes diafragmas para cada rango del espectro para luego reconstruir la señal a base de sumas de los diferentes tramos.

Teniendo en cuenta su uso en el diseño, he escogido un micrófono dinámico por la robustez que presenta que además evitará tener que preocuparse de la alimentación. Además, estos micrófonos funcionan de manera muy similar entre sí, lo que resultará muy conveniente para mantener la flexibilidad del proyecto. El micrófono elegido finalmente será un *MODELO DEL MICRO QUE USE* cortesía del Club Musical Delta. Este se colocará para capturar los sonidos en un pie en la ubicación más adecuada para el instrumento y se conectará a la etapa siguiente ubicada en el suelo mediante un cable.

En cualquier caso, es necesario añadir una etapa posterior de *pre-amplificación o preamp* que adecúe la señal eléctrica del micrófono a la que pueda interpretar la FPGA.

2.2. Preamp

Las principales funciones que va a realizar la etapa de preamplificación será la de amplificar la señal entrante y transformar la señal balanceada proveniente del micrófono en una sin balancear. Se realizará de manera puramente analógica dado que el esquema está muy desarrollado en la literatura sobre el tema. En este caso he implementado el modelo propuesto por P.Allison en [PAm]¹ al que he añadido una fuente de alimentación lineal estándar.

Típicamente, todos los modelos de micrófonos comerciales para aplicaciones de música,

¹Sugerido por el profesor Alfredo Sanz Hervás de la UPM al que agradezco su referencia



Figura 2.2: Saxofón alto

utilizan conexiones balanceadas para proporcionar su señal a la salida. El formato de estos cables de 3 hilos recibe el nombre de *XLR* pero también se pueden utilizar conectores de tipo *Jack de 6.35 mm*² adecuados a este tipo de señales, ilustrados en la figura 2.3.



Figura 2.3: Conector tipo XLR hembra y jack no balanceado macho de 6.35 mm

El fundamento del par balanceado consiste en enviar la señal por dos conductores entrelazados con la polaridad invertida entre sí, envueltos de un tercer conductor que actúa de barrera frente a interferencias electromagnéticas externas. De esta forma, si se produce una interferencia, afectará a ambos conductores de igual manera. Posteriormente, se introduce la señal en destino en un *Amplificador de diferencia*³ que se encargará de amplificar únicamente la diferencia entre ambas señales rechazando la interferencia sufrida. Esta idea muy extendida en los diseños analógicos, permite enviar la señal por largos cables de manera robusta a cambio de introducir un tercer conductor.

Así pues, el diseño de pre-amplificación constará de un amplificador de diferencia mediante amplificador operacional junto con una etapa de amplificación mediante transistores BJT, tal y como se puede ver en el esquema del mismo de la figura 2.4. El potenciómetro permite controlar la ganancia del sistema, es decir, el nivel de señal a la salida del mismo.

2.3. Alimentación

²Este tipo de conector pero sin balancear, es el utilizado en guitarras y bajos eléctricos. Para los auriculares se utiliza jack de 3.5 mm que recibe el nombre de “minijack”.

³Del inglés *Difference Amplifier*, la traducción al castellano puede inducir a error

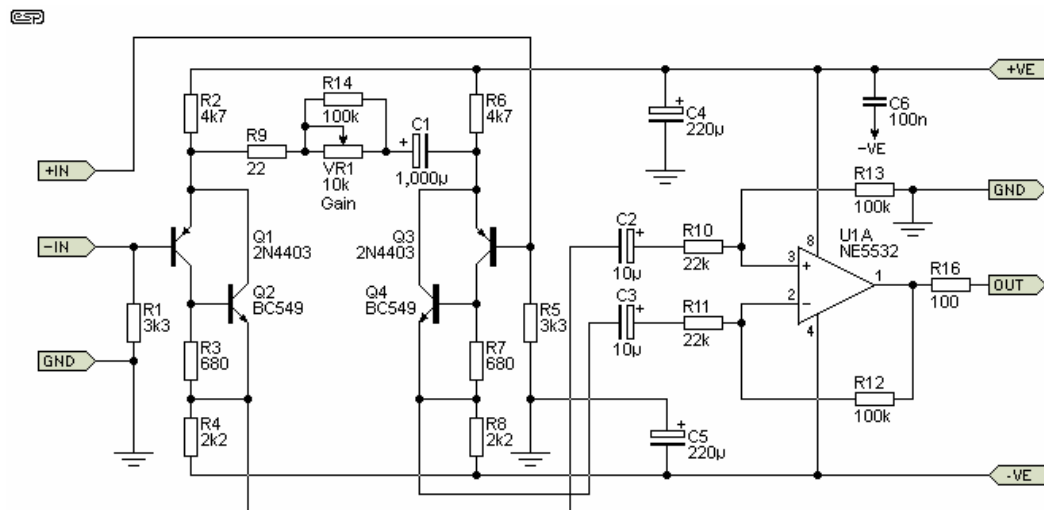


Figura 2.4: Esquema del circuito de pre-amplificación implementado

Capítulo 3

El algoritmo

El propósito de un octavador es proporcionar una señal una octava inferior a la señal introducida. La octava descendente musical equivale a dividir la frecuencia entre 2 por lo que parece que existe una relación evidente con el dominio de Fourier. En el siguiente apartado se discuten diferentes maneras de abordar esta problemática.

3.1. Aproximaciones al problema

Lo primero que pensé cuando abordé el problema fue que se trataba de algo relativamente sencillo: simplemente basta con realizar un diezmado de factor 2 en el dominio de la frecuencia para que al realizar la transformada inversa se obtenga la señal octavada. Evidentemente, esta solución plantea innumerables problemas. En primer lugar, hay que tener en cuenta que las frecuencias del sonido son arbitrarias, por lo que no se puede aplicar ningún tipo de filtro antisolapamiento. En consecuencia, al reconstruir la señal, se produce el mencionado solapamiento, haciendo la señal totalmente irreconocible. Durante todo el siglo XX se han ido desarrollando diferentes técnicas de tratamiento y codificación para la voz, conforme iba la tecnología en aumento. La consecuencia de ello es la aparición de diversos algoritmos que permiten este tipo de operaciones con una carga computacio-

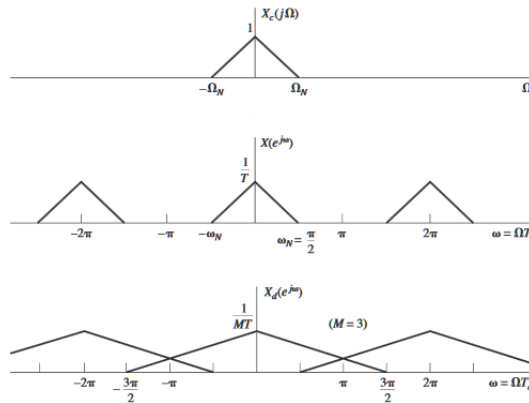


Figura 3.1: Problema de solapamiento tras diezmado

nal relativamente baja. Para este proyecto, se realizará una aproximación de "Vocoder de fase"¹ que se podrá aplicar tanto a la voz como a cualquier instrumento, como se verá más adelante.

3.2. Vocoder

Un Vocoder² es generalmente cualquier aparato que analiza y/o sintetiza la voz humana para lograr algún objetivo concreto, como compresión de datos, multiplexación o encriptación en la mayoría de los casos.

El Vocoder de canal³, desarrollado por los famosos *Bell Labs* en 1928, utilizaba varios filtros multibanda seguidos por detectores de envolvente cuyas señales de control se transmitían al decodificador del receptor. Estas señales de control son mucho más lentas que la señal original a transmitir, por lo que se puede reducir el ancho de banda permitiendo a un mismo medio de transmisión soportar un mayor número de canales, ya sea por radio o cable. Finalmente, el decodificador amplifica estas señales de control y las introduce en los filtros correspondientes a cada banda para poder sintetizar de nuevo la señal original. Además de las ventajas sobre el ancho de banda, también se ayuda a proteger la señal para que no se pueda interceptar. Encriptando las señales de control y modificando los parámetros de los filtros, se puede hacer muy difícil su correcta reinterpretación si no se sincronizan el codificador y el decodificador. Esto popularizó su uso durante la Segunda Guerra Mundial en el bando aliado patentándose diversos diseños.

El concepto se ha mantenido constante durante todo el siglo hasta nuestros días, donde podemos ver implementaciones modernas de la misma idea, por lo que se ha desarrollado una estandarización. La voz humana posee un rango de frecuencias de entre 200 y 3400 Hz típicamente, por lo que se optó por una frecuencia de muestreo de 8 kHz. Es común que se utilice una codificación con 16 bit por muestra por analogía con el estándar CD, pero con utilizar al menos 12 la mayoría de los receptores será capaz reproducir la señal con una fidelidad razonable. Citando un ejemplo, los codificadores según la norma ITU G.729, que son utilizados en telefonía comercial, tienen una buenísima calidad con una tasa binaria de 8 kbps. Actualmente también se utilizan para desarrollar tecnologías relacionadas con la lingüística, la física y la neurociencia.

3.2.1. Vocoder en la música

Paralelamente a su utilización en comunicaciones, el vocoder se comenzó a popularizar durante la década de los 70 como método de síntesis. Cabe mencionar, que durante esta década, surge un gran interés en los músicos por experimentar con diferentes timbres y sonidos en instrumentos conocidos o experimentales. Para aplicaciones musicales, se utiliza una frecuencia portadora proveniente de un instrumento en lugar de extraer la frecuencia fundamental del sonido que se está grabando. El resultado es una deformación del sonido capturado que, por estar afinado en una nota adecuada, produce un resultado agradable al oído. Fue el primer fabricante de sintetizadores y pionero de la música electrónica,

¹Del inglés "phase vocoder"

²Del inglés voice (voz) junto a encoder (codificador)

³Del inglés channel vocoder"

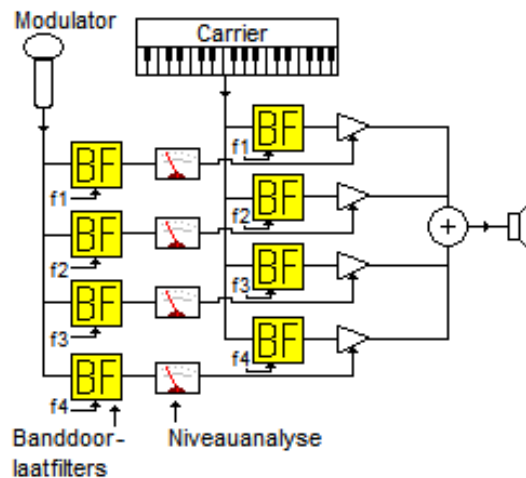


Figura 3.2: Esquema del funcionamiento de un vocoder musical

Robert Moog, el que desarrollo un prototipo llamado *Farad* en 1968 pero no fue hasta 1970 cuando unieron el funcionamiento de esta máquina con el sintetizador modular *Moog* que había lanzado previamente al mercado. Quedaba ya conformada la esencia de utilizar la señal proveniente de un micrófono como moduladora y la proveniente de sintetizador como portadora para modularla. Algunos ejemplos tempranos de músicos reconocidos que utilizaron estos dispositivos fueron Phil Collins, Mike Oldfield, Stevie Wonder, Herbie Hancock o Michael Jackson.

Estos vocoder proporcionaban sonidos a los que el público estaba poco acostumbrado pero que realmente no mantenían una fidelidad tímbrica respecto el sonido que captaban. Por ello se empezaron a utilizar los vocoder de fase los cuales permiten llevar a cabo expansión o compresión en el tiempo y *Pitch Shifting*, es decir, modificar la altura musical del sonido o afinación sin cambiar la forma de onda que proporciona el timbre característico.

El método para hacerlo es el siguiente. En primer lugar se lleva a cabo una transformada mediante STFT (Short Time Fourier Transform) para posteriormente modificar la afinación mediante sub y sobremuestro. Este proceso hace que el audio resultante no resulte reconocible, por lo que es necesario ajustar el valor de la fase de cada muestra para mantener la coherencia entre ellas, de ahí el nombre de vocoder de fase. Una vez calculadas las muestras, se transforman de vuelta al dominio del tiempo, donde se rellena con ceros para obtener la misma duración que la señal entrante. A continuación se explican en detalle estas etapas.

3.3. Transformación a frecuencia: STFT

Una STFT se usa para determinar el módulo y fase de muestras próximas de una señal mientras cambia con el tiempo, haciéndola muy adecuada para aplicaciones en tiempo real. Para ello, se divide la señal en segmentos más cortos de la misma longitud y se calcula la transformada de Fourier de cada uno de ellos por separado. El método para calcular la transformada es indiferente pero al priorizar una baja latencia conviene decantarse por el algoritmo de la Transformada Rápida de Fourier o FFT.

3.3.1. Transformada de Fourier: FFT e iFFT

Para realizar la transformación al dominio de la frecuencia, la opción más adecuada es sin duda el algoritmo de la Transformada Rápida de Fourier o FFT⁴. Este algoritmo calcula la Transformada de Fourier en Tiempo Discreto o DFT⁵ descomponiendo la señal original de longitud N en fragmentos de tamaño $N/2$ como muestra la figura 3.2 y posteriormente multiplicarlo por los términos W_n calculados previamente. Nótese que en los bloques de $N/2$ se puede volver a aplicar el mismo principio de forma recursiva. Esto consigue reducir el tiempo de cálculo porque la transformada propiamente dicha se calcula para una longitud mucho menor. En la ecuación 3.1 correspondiente la DFT genérica podemos ver como la complejidad depende cuadráticamente de la longitud de la entrada $O(n^2)$ mientras que la FFT lo realiza únicamente con $O(n * \log(n))$.

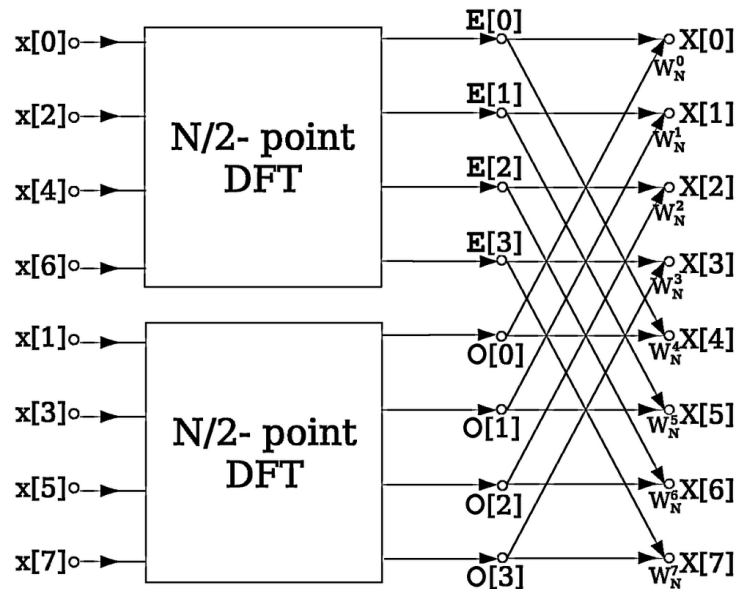


Figura 3.3: Esquema del algoritmo para la realización de la FFT

$$X(k) = \sum_{n=0}^{N-1} x_n e^{-2\pi kni/N} \quad \text{Donde } k = 0, 1, \dots, N-1 \quad (3.1)$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{-2\pi kni/N} \quad \text{Donde } n = 0, 1, \dots, N-1 \quad (3.2)$$

El caso de la transformada inversa es totalmente análogo, el algoritmo de la FFT se puede aplicar de la misma forma para realizar iDFT de forma más rápida, lo que se conoce como iFFT. La ecuación 3.2 muestra la expresión genérica de la iDFT para un señal de N muestras. Cada uno de los parámetros que se utilizan para realizar las transformaciones se encuentra explicado en el apartado de implementación HACER LA CITA.

⁴Del inglés Fast Fourier Transform

⁵Del inglés Discrete Fourier Transform

3.3.2. Solapamiento y enventanado

Dividir la señal entrante en sucesivas tramas es un proceso sencillo, únicamente se almacenan las muestras en una memoria para introducirlas posteriormente en el módulo que realiza la FFT. El problema entonces reside en la propia naturaleza de la FFT y es que esta funciona perfectamente para señales periódicas, pero al dividir la señal, no se garantiza que estas tramas contengan un número entero de periodos. Esto se agudiza especialmente cuando la señal es variante con el tiempo y el número de elementos por trama es independiente de la frecuencia de la señal de entrada.

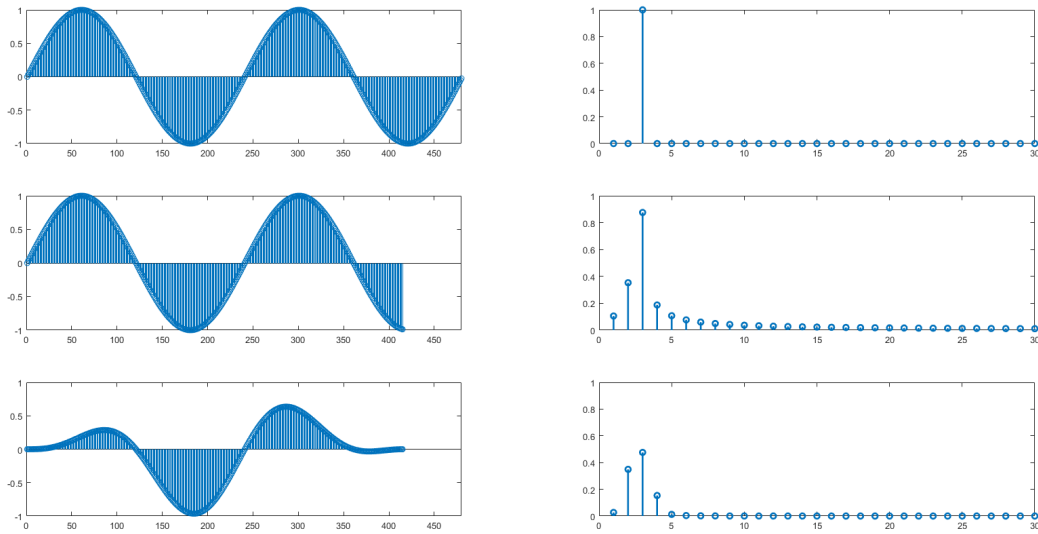


Figura 3.4: Señal sinoidal y su FFT con periodos enteros, recortada y enventanada

En la figura 3.3 se ilustra el problema del recortado arbitrario descrito y su clara mejora al aplicarle enventanado. Frente al caso ideal, el primero, el corte introduce componentes en otras frecuencias que se traducirán como un ruido molesto al final de cada trama al escuchar la transformada inversa. Podemos comprobar como al aplicar una ventana a la señal de entrada, este se hace menor y afecta a menos muestras. Para este ejemplo se ha utilizado una ventana de Hann como la que se utilizará en el prototipo.

Junto al enventanado, se suele aplicar lo que recibe el nombre de *solapamiento*, es decir, en lugar de empezar a construir una trama a continuación de la anterior, la empezamos a llenar antes de que se haya acabado de llenar la trama anterior, repitiendo muestras. De esta forma, las tramas están enlazadas entre ellas evitando una discontinuidad abrupta.

Generalmente se cuantifica este proceso mediante un *factor de solapamiento*, fs , expresado en tanto por ciento. Si una trama t de longitud $n = 100$ muestras tiene un solapamiento de 15 %, las primeras $n * 15 \% = 15$ muestras de t son idénticas a las 15 últimas de la trama anterior, $t-1$, y así sucesivamente.

Lógicamente, no se puede aumentar el factor de solapamiento infinitamente. A partir de un 50 % ya no resulta práctico debido a que para reconstruir la trama t se necesitan muestras de las tramas $t - 1$ y $t + 1$. En este caso, el beneficio de introducir este solapamiento es menor que el coste que hay que pagar: la disminución de la eficiencia del algoritmo. Idealmente realizamos el procesamiento sobre n muestras que luego reconstruimos

pero, al introducir solapamiento, solo un porcentaje de ellas, m , formarán parte de la señal reconstruida. En consecuencia $m < n$ siempre, lo que quiere decir que procesamos más muestras de las que utilizaremos en la reconstrucción: estamos “desperdiciando” recursos.

Este desperdicio de recursos en el procesamiento supone un compromiso doble con las prestaciones. Por un lado, cuanto más disminuya esta eficiencia, más aumentará la latencia, ya que habrá que esperar al cálculo de la siguiente trama para poder finalizar la construcción de la trama presente. Por otro lado, resulta mucho más complejo de cara a la temporización en su implementación.

Como conclusión, debemos elegir un factor de solapamiento $0 < fs < 50$ para que resulte práctico. Tras un modelado en Matlab, he implementado finalmente un valor de $fs = 25\%$ tal y como recomienda Ellis [Ell] en su implementación del vocoder de fase.

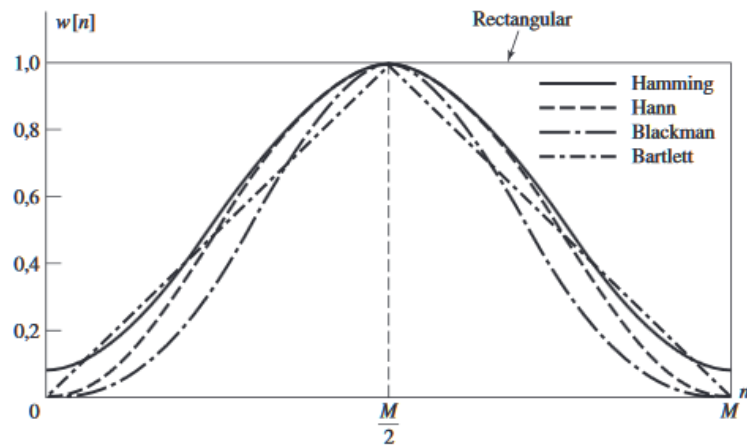


Figura 3.5: Comparativa de las ventanas más utilizadas

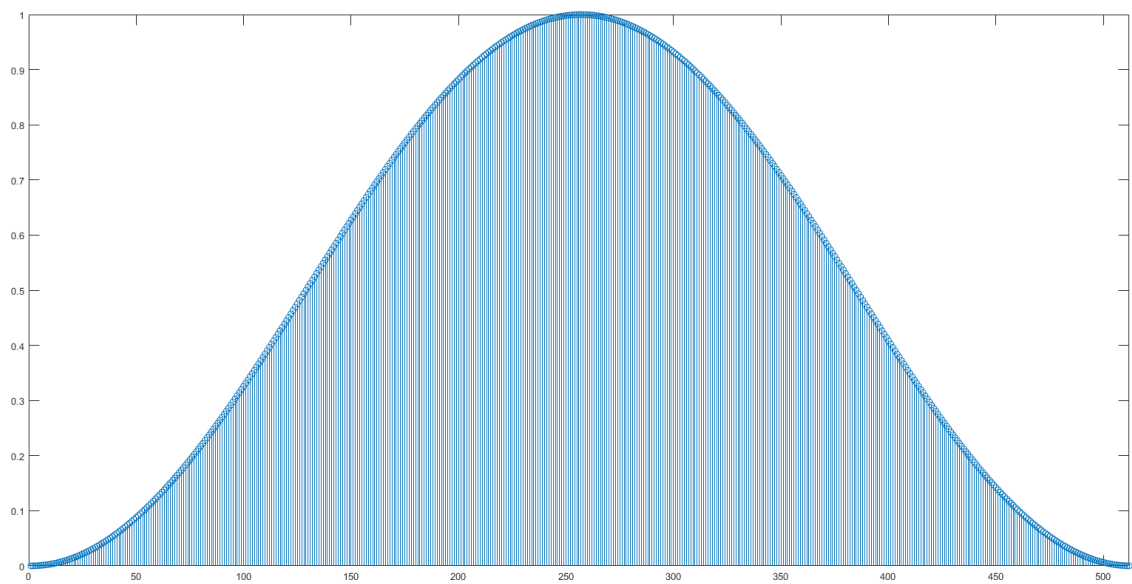


Figura 3.6: Ventana de Hann para $N = 512$ en Matlab

Como ya se ha visto antes, el solapamiento se utilizará junto con un enventanado de Hann cuya expresión se recoge en 3.3. Esta ecuación resulta sencilla de implementar y su uso está muy extendido para aplicaciones de audio en tiempo real frente a algunas de sus alternativas vistas en la figura 3.4. El propio Ellis utiliza en su algoritmo [Ell] una ventana de estas características.

$$H(n) = 0,5 * \left(1 - \cos \left(\frac{2\pi n}{N-1} \right) \right) \quad (3.3)$$

Es evidente, que también se puede aplicar al proceso de la transformada inversa iSTFT. Se utilizará una ventana de las mismas características por las mismas razones antes de introducir las tramas en el módulo que realiza la iFFT. La única salvedad es que para mantener la amplitud de las muestras en la salida igual que las de la entrada, hay que aplicar un factor de escala de $2/3$. En la práctica, lo que se hará será realizar la multiplicación a las constantes de la ventana quedando una ventana de Hann donde cada muestra está escalada por el factor $2/3$, ahorrando así una operación.

3.4. Operaciones sobre la fase

Capítulo 4

Implementación

La implementacion

4.1. Gestión entrada-salida

[h](#)

4.1.1. Pmod i2s2

[h](#)

4.1.2. Consideraciones de implementación

[h](#)

4.2. Controlador de datos

[h](#)

4.2.1. Bancos de memorias

[h](#)

4.2.2. Core FFT

[h](#)

4.3. Controlador global

[h](#)

4.3.1. Displays

h

Capítulo 5

Pruebas y depuración

Pruebas y tal y cual

Capítulo 6

Conclusiones y trabajo futuro

Bibliografía

- [Ell] D.ELLIS, *A Phase Vocoder in Matlab*, LabROSA at Columbia University, Marzo 2003: <http://www.ee.columbia.edu/~dpwe/LabROSA/matlab/pvoc/>
- [Opp] A.V.OPPENHEIM - R.W.SCHAFER, *Discrete-Time Signal Processing*. Pearson Education, 2011
- [Nex] DIGILENT, *Nexys A7 Reference Manual*. <https://reference.digilentinc.com/reference/programmable-logic/nexys-a7/reference-manual>
- [Aud] AUDIO TECHNICA, *What's the pattern?* <https://www.audio-technica.com/cms/site/aa901ccabf1dfc6b/index.html>
- [PAm] P.ALLISON, *Low Noise Balanced Microphone Preamp*. Edited by R.Elliot for Elliot Sound Products ESP, 2008. <http://sound.whsites.net/project66.htm>