UNIVERSIDAD AUTÓNOMA
DEL ESTADO DE MORELOS

FACULTAD DE CIENCIAS QUIMICAS E INGENIERÍA.

"Generador de ondas Kotz para electroestimulación muscular, basado en FPGA"

T E S I S

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

PRESENTA:

ING. COSMOS CLEMENTE CATONGA

DIRECTOR DE TESIS:

DR. J. GUADALUPE VELÁSQUEZ AGUILAR.

CUERNAVACA, MOR. ***JUNIO 2020.***

FORMATO T-2
NOMBRAMIENTO DE COMISIÓN REVISORA
Y DICTAMEN DE REVISIÓN

Cuernavaca, Mor., a 16 de junio de 2020.

DR. LUIS CISNEROS VILLALOBOS
DR. MARIO LIMÓN MENDOZA
DR. J. GUADALUPE VELÁSQUEZ AGUILAR
DR. OUTMANE OUBRAM
DR. MARIO ACOSTA FLORES
PROFESORES DE LA FCQeI
P R E S E N T E

Por este conducto, me permito informarle que ha sido asignado como integrante de la Comisión Revisora de la tesis que presenta el Ing. Cosmos Clemente Catonga, titulada: "Generador de ondas Kotz para electroestimulación muscular, basado en FPGA", realizada bajo la dirección del Dr. J. Guadalupe Velásquez Aguilar en el Programa Educativo de Maestría en Ingeniería Eléctrica y Electrónica. Agradezco su valiosa participación en esta Comisión y quedo a sus órdenes para cualquier aclaración o duda al respecto.

A T E N T A M E N T E
Por una humanidad culta

DRA. VIRIDIANA AYDEÉ LEÓN HERNÁNDEZ
DIRECTORA

D I C T A M E N

DRA. VIRIDIANA AYDEÉ LEÓN HERNÁNDEZ
DIRECTORA DE LA FCQeI

P R E S E N T E

En respuesta a su amable solicitud para formar parte de la Comisión Revisora de la tesis mencionada y una vez realizada la revisión correspondiente, me permito informarle que mi VOTO es:.

DICTAMEN

NOMBRE	VOTO	FIRMA
Dr. Luis Cisneros Villalobos		
Dr. Mario Limón Mendoza		
Dr. J. Guadalupe Velásquez Aguilar		
Dr. Outmane Oubram		
Dr. Mario Acosta Flores		

Se anexan firmas electrónicas

Se expide el presente documento firmado electrónicamente de conformidad con el ACUERDO GENERAL PARA LA CONTINUIDAD DEL FUNCIONAMIENTO DE LA UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS DURANTE LA EMERGENCIA SANITARIA PROVOCADA POR EL VIRUS SARS-COV2 (COVID-19) emitido el 27 de abril del 2020.

El presente documento cuenta con la firma electrónica UAEM del funcionario universitario competente, amparada por un certificado vigente a la fecha de su elaboración y es válido de conformidad con los LINEAMIENTOS EN MATERIA DE FIRMA ELECTRÓNICA PARA LA UNIVERSIDAD AUTÓNOMA DE ESTADO DE MORELOS emitidos el 13 de noviembre del 2019 mediante circular No. 32.

Sello electrónico

VIRIDIANA AYDEE LEON HERNANDEZ | Fecha:2020-06-16 23:04:41 | Firmante

qGi8gfdZP0FUO3k3g7lP2TiHO+RTBQZE2z7B8t08sbOOZGRZKt9FC81E1+gMliQ2WkFb7rXR4r5fCvI4a2p0byzvPWNxZnxOvFILiPOfrYaHueA2+C17xnnAbehzEaVYMsNXMO3EFc0EYUrVVcg+6Y0hbPiS4CukioD7LoDJtBIPmvS2/jCRfxYNZgpyC9rO3cYWZh8Uy/cBncdB0YsRpM0V5vIEtncCMjHGaZ0/pAC3dXkjhlpjzLysWXxCF35vn7qUzPKfQWquiy31/24AjiitRlMzRc+TV8rWeErX16dmEeMJDCXHZxJM3lkwJZN27PDYgngxDMdQCu58zXJhbQ==

Puede verificar la autenticidad del documento en la siguiente dirección electrónica o escaneando el código QR ingresando la siguiente clave:

Vs14e3

https://efirma.uaem.mx/noRepudio/uWGrvLXqKHMYec668qSc0aN0Pr0pqRPY

Se expide el presente documento firmado electrónicamente de conformidad con el ACUERDO GENERAL PARA LA CONTINUIDAD DEL FUNCIONAMIENTO DE LA UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS DURANTE LA EMERGENCIA SANITARIA PROVOCADA POR EL VIRUS SARS-COV2 (COVID-19) emitido el 27 de abril del 2020.

El presente documento cuenta con la firma electrónica UAEM del funcionario universitario competente, amparada por un certificado vigente a la fecha de su elaboración y es válido de conformidad con los LINEAMIENTOS EN MATERIA DE FIRMA ELECTRÓNICA PARA LA UNIVERSIDAD AUTÓNOMA DE ESTADO DE MORELOS emitidos el 13 de noviembre del 2019 mediante circular No. 32.

Sello electrónico

J GUADALUPE VELASQUEZ AGUILAR | Fecha:2020-06-17 14:30:25 | Firmante

Q1tFziZf27o50czZGKqWYaWOzVKsbPvr4tdOiIqiMF4Q7rjRUbNYrARRlJnIbDclw6kNMUqp8YrnuiT4cT3rpkEWrHk7HpVk3q8QTuaipECUCl/gHI+kdAqhQyOfX2/V0pxlE6mJKJ4wtGdu9X7EpoL0OX6ooKs9Ef9Enpbq59cxWNKqGi5Q3WJvJNcnixV1McgDsCyFhSgTU4VHOcN/Qg2Z5eOl561CrB/Xh7qKD7SJwhcYVWBuUayrBCCJtNDqmncukiUFDJy9c+mSFKT/hWW5zfkbBF2vZJ8ADWOhBUgBduSaJM1KLSFotLpKI3/bA0CGqqidC+lFHkaEqQSPMQ==

OUTMANE OUBRAM | Fecha:2020-06-17 14:33:42 | Firmante

aAD+B53zK14ItHhIfWUdRYMzfCmZ+0Lg2gW70iBhy+Jv2+APJKRcWi8AzeQOIypaWancqDllF30kg1RSv7Xb4EWIxfAuE8cDGg7VQpuMTJ5AXEHKWqW2Fg4LDcz3fOf2BcDkjWvLsSkP5uQw83RtUDRcEe6di4bD2aDvgbWoQzYiQY2Baa6j4JWzyFEntiy95ULofaxdsOP/IGmNhgHW0ERrrkEm0fLkkQG3oOhZ1mgE3CMKQNsPwb6XbmqFcYOSbSxUH6iFEYLK12RulSxb0SeeZs3s+AQI+U/T1M7Y5AbN+86mW+LP0vvEXGJAdBsTC4+XFJgQWZCfhhpbJHtF1A==

LUIS CISNEROS VILLALOBOS | Fecha:2020-06-17 15:56:06 | Firmante

k3rkv+8iuFxyb6CUU7dn8Tc1A/jwDiUctsVECA3XD0OW7NJqXFlX0Woj7MJ/syiG5/nwWM5M+bn5sV+Iz7dc74IaYu7vm2D8NM0URrpP8EoWp61dGtgg4KceQCZXj4d4MGlMjj82v13KvoqNvj9moiQ+OB4c3OGZ1WbCkrHHIdd+NRE6iW49VHhk4yUn53jefFz5L33FTjEZ8ztVV5suNhfscsfvGl6jRYQSgEYzALCQKrvwerzpdSJAzVd/2vgEqRo9JY8HTSD4pprFtNGepSatXgmf8ho/Nt48T98q+8vzT5qANczlgU0U0B+rQRFT9U5LmVQcM6nnopuly24pdg==

MARIO LIMON MENDOZA | Fecha:2020-06-17 17:17:45 | Firmante

hYt7NvsyvSEOKvEFnOxe46CaayUYeIl88sWNjojDJx2o1wucswD/oGg9KWLraKgsZbW+Y6eqyM1ybfY+k5lp4P16N4qqSwAjCyYNMX+fuB/cmXZUjEcIBRKjVlXf5TyJrG7VZS3rKZxiZenBWnJ0Ty/CCnbcDMpYm/+wf2qxJx/2gWGYXqDlOiF3c8vyjEhDEWTuSgr2jQr3Hv+ZwBHUkJpNyY4jsYiDWdu7c2BSFAjRjYIctPibXN3lLE8Z+LlfDsTM18MrB1Rh5TaRZYXWcYQDkO6U654kXZ9zxLtLncM6U/Dki2vkWoyT4P0r2H/uR9xUsYjuQyYirHoVqaInVA==

MARIO ACOSTA FLORES | Fecha:2020-06-17 21:38:22 | Firmante

cDbYif4Wm88GWWdLwvXC9s4XrZIkS9rCxIBTyDk/NvgAZiEiMerVc6fave0370XK54PWhuG67YE7csiVKkeB16PUJVmdnyh4PeYIo+4pzLxv4mD22kn/lsDqGIzGN+YlHjIoqvF6ETP5nKTc5c8yFnMizpNAwTdDT9SYiqQrlYcVGrbGCC42ifro398nXUH8welQ/l6ByAyxMhB/k6Kpo7yoUxADKAyq+6TSHGS940ikFAqNSIbSRde48ccRmfS9ii/AVtfiHzjnJiHl+SoDeKxki+AnPX6tTbX3m/FccO4Xr1pBDvJzKiLEbeqbJDLYMpBbdWKUm7frEGLP72kyHQ==

Puede verificar la autenticidad del documento en la siguiente dirección electrónica o escaneando el código QR ingresando la siguiente clave:

UhH8d0

https://efirma.uaem.mx/noRepudio/pOaYb2dnhUA618H3PuGomo1Zq6GjMqlz

Dedicada

A mis padres:

Tirso Clemente Jiménez y Micaela Catonga Xopo

A mis hermanos:

Universo, Tirzo, Micaela, Azucena y Esmeralda

A mi pareja:

Rosalinda, por su apoyo incondicional y verdadero, porque a pesar de las dificultades me apoya en mis sueños y metas.

Agradecimientos

Mi profundo agradecimiento a la **Facultad de Ciencias Químicas e Ingeniería**, de la UAEM; por confiar en mí y permitirme realizar parte del proceso investigativo dentro de su establecimiento educativo.

A mis profesores en especial al **Dr. Luis Cisneros Villalobos, Dr. Mario Limón Mendoza, Dr. Outmane Oubram, Dr. Mario Acosta Flores, Dra. Viridiana A. León Hernández, Dr. Roy López Sesenes**, quienes con la enseñanza de sus valiosos conocimientos hicieron que pueda crecer día a día como profesional, gracias a cada una de ustedes por su paciencia, dedicación, apoyo incondicional y amistad.

A mis **padres y hermanos**, quienes son mi motor y mi mayor inspiración, que a través de su amor, paciencia, buenos valores, ayudan a trazar mi camino y alcanzar mis objetivos.

No puedo dejar de agradecer especialmente a **Rosalinda**, en el pasado mi compañera fiel de proyectos culturales, y ahora de corazón y vida.

Finalmente quiero expresar mi más grande y sincero agradecimiento al **Dr. J. Guadalupe Velásquez Aguilar**, director de tesis durante todo este proceso, quien con su dirección, conocimiento y enseñanza se logró el desarrollo de este trabajo. Por su infinita y grandiosa amistad.

Resumen

En este trabajo, se desarrolló un generador de ondas Kotz (ondas rusas), que se puede utilizar en la electroterapia de estimulación a frecuencia media. Para la realización del sistema, se hizo una revisión de los conceptos elementales sobre las señales eléctricas y la forma en que estas funcionan dentro del cuerpo humano. Los tipos de corrientes eléctricas y algunas de sus propiedades específicas, son analizadas para identificar los efectos terapéuticos que pueden proporcionar al aplicarse al cuerpo humano, en particular las ondas rusas.

Las señales generadas por el sistema son diseñadas utilizando modelos matemáticos, tanto para la señal base (frecuencia constante) como para la señal moduladora (esquema de tratamiento en base a tiempo, frecuencia y amplitud de la onda). Los datos obtenidos del modelado son programados posteriormente en dispositivos electrónicos digitales reconfigurables, en este caso en una matriz de puertas lógicas programables en campo (Field-Programmable Gate Array, FPGA).

Finalmente, el sistema en físico utiliza una tarjeta basada en FPGA modelo DE10-Lite. La arquitectura de hardware se describe utilizando el VHDL (Lenguaje de Descripción de Hardware). El sistema es validado mediante un banco de pruebas, generando diferentes señales y observando su estabilidad en los períodos de tiempo de acuerdo con los utilizados en las terapias de rehabilitación.

Abstract

In this work, a Kotz wave generator (Russian waves) was developed, which can be used in medium frequency stimulation electrotherapy. To carry out the system, a review of the basic concepts of electrical signals and the way they work within the human body was performed. The types of electric currents and some of their specific properties are analyzed to identify the therapeutic effects they can provide when applied to the human body, in particular Russian waves.

The signals generated by the system are designed using mathematical models, both for the base signal (constant frequency) and for the modulation signal (treatment scheme based on time, frequency and wave amplitude). The data obtained from the modeling is subsequently programmed into reconfigurable digital electronic devices, in this case in a field programmable gate matrix (FPGA) matrix.

Finally, the physical system uses a DE10-Lite FPGA-based card model. Hardware architecture is described using VHDL (Hardware Description Language). The system is validated by means of a test bench, which generates different signals and observes its stability over time in accordance with those used in rehabilitation therapies.

ÍNDICE GENERAL

Dedicatorias ii
Agradecimientos iii
Resumen iv
Abstract v
Índice General vi
Lista de Figuras ix
Lista de Tablas x
Nomenclatura xi
Simbología xiii

1. Introducción 1
1.1 Antecedentes 2
1.2 Planteamiento del Problema 3
1.3 Objetivos 4
1.3.1 Objetivos específicos 5
1.4 Justificación 5
1.5 Metodología para el desarrollo del proyecto 6
1.6 Limitaciones 6
1.7 Estructura de la Tesis 7

2. Fundamentos de Corrientes Rusas 8
2.1 Antecedentes 9
2.1.1 Revisión bibliográfica 10
2.2 Corriente eléctrica en electroterapia 12
2.2.1 Aspectos fisiológicos de la corriente eléctrica 12
2.2.1.1 Fenómenos no nocivos causados por la electricidad en el ser humano 13
2.2.2 Tipos de corriente eléctrica 14
2.2.2.1 Corriente eléctrica en estado constante 14
2.2.2.1.1 Galvanización 14

2.2.2.1.2 Ionización medicamentosa 15
2.2.2.1.3 Parámetros de la corriente constante 17
2.2.2.2 Corriente eléctrica en estado variable 18
2.2.2.2.1 Tipos de corrientes variables 18
2.2.2.2.2 Parámetros de la corriente variable 19
2.2.2.2.3 Corrientes Interferenciales 20
2.3 Modelos matemáticos para generación de barridos 21
2.3.1 Modelo matemático de la función trapezoidal 21
2.3.2 Modelo matemático de la función cuadrada 23
2.3.3 Serie de Fourier Trigonométrica 23
2.3.3.1 Función trapezoidal para barridos 23
2.3.3.2 Función cuadrada para señales interferenciales 25
2.4 Onda Rusa portadora de señal 25
2.4.1 Modulación en Amplitud 27

3. Tecnologías para diseño de sistemas empotrados 29
3.1 Antecedentes 30
3.2 Factores a considerar en el desarrollo de aplicaciones 30
3.3 Tecnologías digitales para diseño de sistemas empotrados 31
3.3.1 Microprocesador (MPU) 31
3.3.2 Microcontrolador (MCU) 32
3.3.3 Procesador Digital de Señales (DSP) 33
3.3.4 Dispositivos reconfigurables (FPGA) 35
3.3.5 Circuitos integrados de aplicación específica (ASIC) 36
3.4 Análisis de la tecnologías para aplicaciones en tiempo real 36
3.4.1 Discusión de las alternativas 37
3.4.1.1 MPU 37
3.4.1.2 MCU 38
3.4.1.3 DSP 39
3.4.1.4 FPGA 39
3.4.1.5 ASIC 40
3.5 Elección: DSP vs FPGA 40

4. Generación de barridos para ondas interferenciales 43
4.1 Efectos de las frecuencias en el músculo 44
4.2 Parámetros de los barridos 45
4.3 Generación de barridos utilizando MatLab 46
4.4 Arquitectura para la generación de barridos hardware 47
4.5 Generación de interfaz software para el control de los barridos 49

5. Conclusiones 52
5.1 Discusión. 53
5.2 Contribuciones del Trabajo 54
5.3 Conclusión 54
5.4 Trabajo Futuro 55

Apéndice A: Códigos de simulaciones e implementaciones en hardware 56
Bibliografía 63

LISTA DE FIGURAS

Figura 2.1 Densidad del flujo de la corriente eléctrica 15
Figura 2.2 Clasificación de corrientes variables 18
Figura 2.3 Forma de impulsos aislados utilizados en electroterapia 19
Figura 2.4 Trenes de impulsos no modulados 20
Figura 2.5 Curva envolvente de frecuencia media 21
Figura 2.6 Forma del trapecio y altura máxima con los tiempos como parámetros de control 22
Figura 2.7 Forma de onda cuadrada y las características de la señal inferencial 23
Figura 2.8 Forma de onda rusa y sus diferentes fases 26
Figura 2.9 Señales base para ondas rusas 27
Figura 2.10 Onda Kotz modulada con función trapezoidal 28
Figura 3.1 Diagrama a bloques de un microprocesador Intel Pentium 32
Figura 3.2 Diagrama a bloques de un microcontrolador PIC10F200 de Microchip 33
Figura 3.3 Diagrama a bloques de un DSP TMS320C6211 de Texas Instruments 34
Figura 3.4 Elementos de la lógica reconfigurable 35
Figura 4.1 Contornos para modulación y barridos de las ondas Rusas 45
Figura 4.2 Diagrama de flujo del algoritmo para generación de barridos 46
Figura 4.3 Acondicionamiento de la señal portadora 46
Figura 4.4 Barridos generados en base a modulación 47
Figura 4.5 Entorno de programación de dispositivos FPGA 48
Figura 4.6 Corrientes interferenciales con ondas rusas 49
Figura 4.7 Entorno de programación Qsys 50
Figura 4.8 Entorno de programación con lenguaje de alto nivel 50
Figura 4.9 Interfaz gráfica realizada en Java para control de la arquitectura 51

LISTA DE TABLAS

Tabla 2.1 Intensidades de la iontoforesis recomendadas con base a superficie corporal 16
Tabla 3.1 Análisis de tecnologías digitales 37

NOMENCLATURA

Notación	Significado	Descripción
Arduino	Plataforma de hardware y software libre	Placa con microcontrolador y entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinares.
MPU	Microprocesador	Circuito integrado que actúa como la Unidad Central de Proceso de una computadora
MCU	Microcontrolador	Circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria.
FPGA	Matriz de Puertas Lógicas Programables en campo (Field-Programmable Gate Array)	Dispositivo programable que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada en el momento mediante un lenguaje de descripción especializado.
DSP	Procesador Digital de Señales	Circuito integrado basado en un procesador o microprocesador que posee un conjunto de instrucciones, hardware y software optimizados para aplicaciones que requieran operaciones numéricas a muy alta velocidad.
ASIC	Circuito Integrado de Aplicación Específica	es un circuito integrado hecho a la medida para un uso en particular
DDR3	Double Data Rate type three Synchronous Dynamic Random-Access Memory	Tipo de memoria RAM
CPU	Unidad Central de Procesamiento	Núcleo de un procesador que interpreta las instrucciones de un programa informático mediante la realización de las operaciones básicas aritméticas, lógicas y de entrada/salida del sistema
USB	Universal Serial Bus	Bus de comunicaciones que sigue un estándar que define los cables, conectores y protocolos usados en un bus para conectar,

		comunicar y proveer de alimentación eléctrica entre computadoras, periféricos y dispositivos electrónicos
FPU	Floating Point Unit	Unidad para procesamiento de datos en punto flotante
RAM	Random Access Memory	Memoria de acceso aleatorio que permite almacenar y extraer información de ella.
HDL	Lenguaje de Descripción de Hardware (Hardware Description Language)	Lenguaje de programación para modelar la arquitectura y comportamiento de sistemas electrónicos discretos.
IEEE	Organización internacional para estándares eléctricos y electrónicos	Fomenta la innovación tecnológica y la excelencia para el beneficio de la humanidad. Una de sus actividades es normalizar símbolos, diagramas, esquemas eléctricos y electrónicos para el uso estandarizado
MATLAB	Laboratorio de Matrices (MATrix LABoratory)	Herramienta de software matemático que ofrece un entorno de desarrollo integrado con un lenguaje de programación propio. Se comunica con otros lenguajes de programación y hardware
Quartus II	Nombre del software para programación	Herramienta de software producida por la empresa Intel FPGA (anteriormente Altera) para el análisis y la síntesis de diseños, análisis temporales y configuración de dispositivos digitales
VHDL	Acrónimo para el Lenguaje de Descripción de Circuitos Electrónicos Digitales	Herramienta de software que permite describir circuitos electrónicos puramente digitales, síncronos y asíncronos de alta velocidad, con base a los estándares de las normas de la IEEE.
EEM	Electroestimulación	Técnica que se utiliza como terapia física para conseguir una contracción del musculo a tratar, utilizando impulsos eléctricos
pH	Potencial de hidrógeno	Medida para determinar la acidez o alcalinidad de una solución

SIMBOLOGÍA

Se incluyen símbolos generales (con letras latinas y griegas), subíndices, superíndices y abreviaturas. Cada una de estas listas esta´ ubicada en orden alfabético de acuerdo con la primera letra del símbolo, con base al Sistema Internacional (SI).

Símbolos con letras latinas

Símbolo	Término	Unidad SI	Definición
I	Intensidad de corriente eléctrica	A	Amperes
V	Tensión eléctrica	V	Volts
f	Frecuencia	Hz	Hertz
f_{Kotz}	Frecuencia de Kotz	Hz	Hertz
A	Área	m^2	$\int\int dxdy$

Símbolos con letras griegas

Símbolo	Término	Definición
$\sum$	Sigma	Sumatoria
A	Área	m^2
Ω	Ohm	Resistencia
μ	Micro	10e-6

Capítulo 1
Introducción.

En este capítulo se describen en forma general los pormenores de la electro-estimulación, sus usos en la electroterapia y algunos antecedentes que incentivaron el uso de estas técnicas en diferentes áreas, entre ellas la analgesia. Se define el problema a resolver y se describen también los objetivos, la justificación, la metodología utilizada y limitaciones del sistema, así como una breve descripción de la información que se encontrará en este documento de tesis.

1.1 Antecedentes

La técnica de electroestimulación se ha utilizado desde hace varias décadas en la medicina, sus usos se extienden desde fines estéticos hasta terapéuticos.

En el ámbito de rehabilitación, está la técnica se emplea: para el fortalecimiento muscular; facilitar la contracción muscular voluntaria insuficiente; reeducación muscular; prevención y el tratamiento de la atrofia, debido a las cirugías o inmovilización; mantenimiento de rango de movimiento; manejo de espasticidad (trastorno motor del sistema nervioso en el que algunos músculos se mantienen permanentemente contraídos); como técnica asistencial para la recuperación de la función muscular después de lesiones ortopédicas, reducción de contracturas y edema.

La reducción de la actividad y los fenómenos de inhibición tiene como efecto las lesiones del aparato locomotor, acompañada generalmente en la disminución de fuerza y con ella el aumento de la fatiga muscular. Aunque no todas corresponden la misma problemática en los músculos duros y blandos, la electroestimulación suele aplicarse para tratamientos en los cuádriceps, espalda, piernas, lumbalgia, ciática y los casos de la migraña, principalmente.

El Dr. Yakov Kotz, médico ruso, en una conferencia de 1977, dio a conocer el descubrimiento de una onda eléctrica muy particular con el cual realizó experimentos y como resultado, obtuvo el incremento de fuerza muscular en atletas rusos de hasta un 40%. Los tratamientos originalmente fueron considerados con frecuencias bajas y altas, con el objetivo de encontrar una zona en particular del espectro electromagnético para ser usadas con fines óptimos. En tanto también se dieron resultados con base a experimentos con humanos en diferentes días y modos de electroestimulación muscular, en la mayoría de los casos de las extremidades (brazos y piernas). Sin embargo el Dr. Kotz no profundizó en una modulación particular para acelerar resultados, pero sí de mantener sin deformidad la onda con amplitud y frecuencia constante. Con el tiempo los parámetros se fueron identificando y estandarizando, cada vez se consideraron otras formas de onda para la terapia física, al grado de diseñar equipos más complejos y corrientes interferenciales completamente moduladas, para cada tratamiento.

Por lo anterior, este trabajo está enfocado en la generación de ondas interferenciales, en la gama de media frecuencia comprendida de 2,000Hz - 10,000Hz; particularmente en 2,500Hz, para el área de la electroterapia, para lo cual se utilizarán dispositivos digitales programables como los son los FPGA, con el objetivo de reducir el costo respecto a los que existen en el mercado.

1.2 Planteamiento del problema

Se estima que a nivel internacional entre el 10% y 55% de la gente adulta padece algún dolor agudo o crónico [1]. En México el valor porcentual estimado es de 27%, con base a los estudios de investigadores del Instituto Nacional de Ciencias Médicas y Nutrición "Salvador Zubirán" [2]. Las zonas donde se identifican los dolores musculares más comunes son en la ciática, las cervicales, la espalda, los hombros, artrosis en las manos y rodillas.

Es muy poca la población que tiene el conocimiento de la tecnología de instrumentación electrónica que se utiliza para atenuar y eliminar patologías de origen muscular y circulatorio, muchas veces sin provocar daños en el organismo. Esto le ocurre también a las instituciones privadas y públicas como la Secretaría de Salud de México.

Existen en el mercado especializado una variedad de equipos electroestimuladores para estos propósitos, sin embargo, en general sus precios son muy altos y la mayoría de la población, que tiene conocimiento de ellos, no tiene los medios económicos para adquirirlos. Los electroestimuladores existentes suelen llevar por implícito un estilo perteneciente a los hábitos y conveniencia de cada zona o país de procedencia. Por ejemplo, en los equipos desarrollados en diferentes países: España, Francia, Holanda, Italia, Alemania, Suiza, Estados Unidos y orientales (como China, Taiwan o Korea) se aprecian diferencias significativas entre ellos.

También se detectan matices comunes de procedencia del mismo lugar. En general, no se tiene la facilidad para modificar la señal controladora de las ráfagas, ya que solamente permiten tratamientos predeterminados con pocas opciones ya programadas para el uso terapéutico. Aunado a esto, en muchos casos la fabricación de los equipos tiene mucha aplicación con fines estéticos y deportivos.

Por lo anterior, ¿Por qué no diseñar un algoritmo con lenguaje de programación computacional donde realice cálculos que nos permita crear las Ondas Kotz, los parámetros fundamentales para cada tratamiento, con los requisitos establecidos y ser reproducidos en sistemas digitales involucrando los rangos de las señales; para posteriormente ser utilizados en la electroestimulación muscular de alta y baja frecuencia, como tratamiento en analgesia eléctrica; así como también modificar los barridos rápidos y lentos en el batido de señales eléctricas para el campo de la investigación médica e industrialización de equipos médicos en un futuro cercano?.

Este proyecto se enfoca a crear señales sinusoidales que serán moduladas de forma trapezoidal respecto al tiempo, para que actúen como controlador de barridos en tratamientos de alta y baja frecuencia, esto es, la señal sinusoidal tendrá una frecuencia constante y magnitud variable comportándose como ráfagas. Esto con base a tratamientos establecidos bajo normas internacionales.

Mediante el desarrollo de programas software, es posible generar secuencias con diferentes tiempos para dichas ráfagas y barridos, con la finalidad de mantener controladas las Ondas Kotz y así facilitar su aplicación en áreas dedicadas a la electroterapia y en la investigación médica. Esta tarea implica el almacenamiento de los datos obtenidos en el software, en medios digitales como pueden ser los dispositivos microcontroladores (PIC's, Arduino, MSP), o tarjetas FPGA's, donde las corrientes interferenciales serán generadas de forma eficiente y con un ruido tolerable y previamente establecido.

1.3 Objetivos

1.3.1 Objetivo general

Generar señales eléctricas-electrónicas con parámetros de las Ondas Rusas a través de software, las cuales estén en un rango de comportamientos interferenciales, que serán reproducidas con sistemas digitales y una interfaz de control sencilla, y utilizadas en pruebas y ensayos, con barridos de alta y baja frecuencia, para la identificación de parámetros en el tratamiento muscular o tejidos biológicos.

1.3.2 Objetivos específicos

1. Desarrollar un programa computacional que permita el cálculo y la simulación gráfica de los parámetros de alta y baja frecuencia, de una función trapezoidal que muestren los barridos planeados y elegidos por el usuario con una interfaz sencilla.

2. Utilizar los valores teóricos cuantizados para ser reproducidos en medios o dispositivos digitales, como pueden ser tarjetas FPGA's, Arduino o PIC's; así como cualquier otro hardware de bajo costo y de mayor circulación comercial para ser adquirida.

3. Crear una interfaz gráfica donde se muestren los eventos en tiempo real durante los barridos de alta y baja frecuencia, por cada tratamiento de las Ondas Rusas elegidas.

1.4 Justificación

El desarrollo de las Ondas Rusas con una amplia versatilidad para el control de barridos con baja y alta frecuencia en base a un dispositivo con arquitectura abierta es de gran relevancia, dado que muchas de las empresas dedicadas a la venta de estos instrumentos de electroestimulación, tienen costos muy elevados. Es por esta razón que muchas clínicas y hospitales tanto públicos como privados mantienen un presupuesto muy limitado que los hace prescindir de estos dispositivos.

La instrumentación médica debe estar certificada por estándares y normas estrictas, que permitan una confiabilidad para ser consideradas en experimentos muy específicos. Por mencionar algunos ejemplos, barridos lentos o rápidos para estudios de la piel, simulación de comportamiento muscular biomecánico, analgesia eléctrica, tratamiento de patologías con origen muscular y circulatorio; u otros padecimientos similares.

El desarrollar tecnología para este tipo de tratamientos, facilita el tener disponibilidad y flexibilidad en equipo especializado para las áreas de la electrofisiología, instrumentación médica y aquellas que necesiten ser considerados para estudios, ensayos y experimentos específicos para el desarrollo de las ciencias principalmente médicas.

1.5 Metodología para el desarrollo del proyecto

Dado que este proyecto involucra la obtención de parámetros utilizando ecuaciones matemáticas para posteriormente ser implementadas en dispositivos hardware, se seguirá la siguiente metodología en la realización:

1. Modelar el funcionamiento de las corrientes interferenciales a través de la Serie de Fourier Trigonométrica, para la adquisición de datos en barridos lentos y rápidos, con base a las metodologías de la modulación en amplitud.
2. Implementar códigos numéricos en lenguajes de programación y descripción para dispositivos puramente digitales.
3. Verificar los resultados obtenidos con la literatura especializada de la electroestimulación y las áreas similares.
4. Diseñar un módulo de potencia portátil, que permita acoplarse a otro módulo externo para reforzar la tensión eléctrica y el flujo de corriente.

1.6 Limitaciones

Dado que acorde a las normas establecidas en esta área, en los experimentos físicos no se permiten las pruebas directas en humanos, tejidos In vivo e In situ, solo se verificará la funcionalidad del sistema, comparado con equipos comerciales.

Los materiales empleados en la construcción del sistema son de uso comercial; con la finalidad de que este pueda ser utilizado para fines académicos y de investigación, pero servirán como base para el diseño de un prototipo especializado.

1.7 Estructura de la Tesis.

La tesis está dividida en tres partes: la primera parte es una revisión de los antecedentes y conceptos básicos de electroterapia, electroestimulación, tipos de corrientes utilizadas y los efectos que estas tienen en el cuerpo humano (Capítulo 2). También se describen las generalidades y herramientas para el diseño con hardware reconfigurable. Además, se analizan las ventajas que ofrece la implementación de algoritmos en este tipo de sistemas, así como las consideraciones generales que aseguren un buen compromiso entre consumo de recursos hardware y precisión de los resultados obtenidos (Capítulo 3).

La segunda parte trata sobre el desarrollo de algoritmos para generar señales utilizando análisis frecuencial así como aspectos de modulación para la generación de barridos (Capítulo 4). Se utiliza el software MatLab como herramienta computacional de apoyo. Los resultados obtenidos serán acondicionados para su utilización en el desarrollo de arquitecturas hardware basadas en FPGA's que generen las señales físicas.

En la tercera parte se describe la arquitectura hardware propuesta para el sistema. Se realiza una evaluación cuantitativa del sistema, provocado por la limitación en el formato de datos utilizado. Se analiza el rendimiento y la eficiencia final del sistema desarrollado en el FPGA, comparándolo con los resultados obtenidos en simulación a través de la herramienta computacional MatLab (Capítulo 5).

En base a los resultados obtenidos, se realiza una discusión sobre la flexibilidad del sistema, limitaciones y posibles mejoras para complementarlo.

Capítulo 2.

Fundamentos de Corrientes Rusas.

La energía eléctrica ha sido utilizada desde hace un siglo y medio en la fisiología y patología, en esta última teniendo un interés muy particular para sus aplicaciones terapéuticas. Se le ha dado usos en las diversas y múltiples variaciones de la cual conforman diferentes tipos de flujo de corrientes eléctricas. Las corrientes eléctricas tienen características dependiendo de sus formas, estás se definen por parámetros como: la amplitud, la frecuencia, la dirección, tiempo de aplicación y esquema de modulación, lo cual confiere muchas propiedades biológicas muy específicas. En este capítulo se revisan los conceptos elementales sobre la electricidad y los efectos que esta tiene sobre el cuerpo humano. Se describen los diversos tipos de corrientes, sus propiedades específicas, así como las cualidades terapéuticas al ser aplicadas al cuerpo humano, en específico de las corrientes interferenciales.

2.1 Antecedentes.

La electroterapia considera una parte de la energía del espectro electromagnético para fines de aplicación en el organismo humano, particularmente en los tejidos. Las respuestas que generan la aplicación de estas señales pueden favorecer la rehabilitación y corrección de males circulatorios en zonas de músculo duro.

La electroestimulación es una técnica utilizada hoy en día en terapias para los tratamientos de patologías cuya evolución implica pérdida de fuerza muscular. Su aplicación en esos casos es un tratamiento funcional del tejido muscular, es decir, en las células de la misma especie contráctiles que consiguen el correcto funcionamiento mecánico a partir de la energía química, con la interacción de las proteínas actina y miosina fibrosa. La miosina fibrosa tiene filamentos con una longitud uniforme de 1.6μm y un diámetro de 15nm, que conjuntamente con la actina, permiten principalmente la contracción de los músculos e interviene en la división celular y el transporte de vesículas. La vesícula es un orgánulo formado por un compartimento pequeño y cerrado, este almacena, transporta o digiere productos y residuos celulares y son una herramienta fundamental de la célula para la organización del metabolismo.

Acorde a estudios realizados por el profesor Kotz en 1977, es posible obtener un incremento de fuerza del músculo tratado, es por eso, que paso al entrenamiento del mundo deportivo. La Onda Rusa estudiada es una señal con una frecuencia de 2.5kHz, con la características de una función senoidal bifásica y simétrica, tensión eléctrica de $12V_{pp} - 100V_{pp}$ y una intensidad de flujo de corriente de $0.001\ A/cm^2 - 0.06\ A/cm^2$, generalmente.

La frecuencia de 2.5kHz posibilita una disminución de la impedancia o resistencia de piel, para lograr así una menor sensibilidad al paso de las ondas eléctricas donde se permite usar la mayor tensión eléctrica e intensidad de corriente sin ser percibidas. Existen dos técnicas fundamentales y básicas para la electroterapia con dichos fines:

- *Alta frecuencia y baja intensidad*: se aplica utilizando barridos con frecuencias comprendidas entre 50Hz y 150Hz, este último es el más adecuado en tratamientos prolongados, el efecto es más rápido pero de corta duración.

- *Baja frecuencia y alta intensidad*: se aplica utilizando barridos con frecuencias comprendidas entre 0.1Hz y 5Hz de manera indistinta. El efecto es lento, pero es más duradero.

El tipo de corriente que genera un aparato electroestimulador conlleva impulsos negativos y positivos, es decir, cambia la polaridad, de esta forma compensa los efectos polares, evitando así las quemaduras químicas que tan comunes eran en el pasado. Por lo anterior, se debe considerar el modo de colocación del electrodo, ya que no corresponde al tipo de intensidades de corrientes que se usan en la actualidad.

Hoy en día existen muchos tipos de electroestimuladores, hay equipos de Estimulación Eléctrica Transcutánea Nerviosa (TENS), equipos para electroestimulación en estética (como para el tratamiento de la celulitis y otras), equipos específicos para el deporte, y hay aparatos que reúnen todas las posibilidades. Obviamente con propiedades y control de las señales muy poco consideradas.

2.1.1 Revisión bibliográfica

El desarrollo de un electroestimulador requiere de una familiarización teórico-experimental, para lo cual se realizó una inspección de la teoría y desarrollos tecnológicos donde se han utilizado Ondas Kots. Los artículos son recomendados por instituciones de prestigio y especialistas en el área como los son: periódico de Terapia Ortopédica y Deporte Físico, la base de datos de la Asociación Americana de Terapia Física (APTA) y el Centro Nacional de Información en Biotecnología (NCBI). También se incluyeron otros artículos como la Enciclopedia Médico-Quirúrgica, Manual del Electroestimulador ECAM y el Bimodal TENS. De esta revisión se encontró lo siguiente:

R. P. Walmsley y G. Letts, J. Vooys, [3] llevan a cabo una comparación de la contracción muscular con dos experimentos para pruebas de corrientes senoidales en una gama de frecuencias 1.6kHz − 2.5kHz con ráfagas de 100Hz simétricos. Usando equipos para simular las Ondas Rusas, Electrosim 180, Micromed Instruments Inc, Montreal, Quebec. Máquina interferencial de media frecuencia Nemectrodyne 8, Nemectron Co. Germany, en la gama de 4kHz−4.1kHz con

ráfagas de 75Hz. Concluyendo que la región de mayor contracción muscular es la media frecuencia con resultados de 10%− 30% satisfactorios.

A. Chantraine, C. Gobelet y J. L. Ziltener, [4] hacen una reseña de los diferentes campos de la electroterapia, conceptos elementales acerca de la electricidad y su acción en los distintos tejidos del cuerpo humano. Describen los tipos de corrientes y sus propiedades específicas, efectos terapéuticos, indicaciones y contraindicaciones de la electroestimulación en baja, media y alta frecuencia en el músculo patológico o sano.

Ricardo Ávila Araya [5], puntualiza aspectos importantes de las diferentes frecuencias como ráfagas para la rehabilitación de músculos desnervados o debilitados, entre 1Hz − 150Hz, Así como también la descripción de la modulación en amplitud de las Ondas Rusas, con los efectos de las distintas frecuencias como corrientes interferenciales en 2.5kHz.

Alex R. Ward y Nataliya Shkuratova [6] realizan una revisión detallada de los trabajos originales del Dr. Kotz y compañeros de trabajo, como Andrianova que concluye el uso de la onda senoidal con el rango de 1kHz − 2.5kHz y ráfagas de 50Hz, definiendo como región óptima 2.5kHz. Para Xvilion reportó no usando la frecuencia de 2.5kHz, si no con una señal rectangular o pulsos de trenes de 1kHz y ráfagas de 50Hz definiendo el protocolo de tratamiento 10/50/10; que consiste en 10s de electroestimulación 50s de reposo con una duración de la sesión de 10min totales, durante 9 − 19 días consecutivos. Donde ambos colaboradores realizan estudios con atletas de edades 15 − 17 años, formando 4 grupos, del cual obtienen resultados de fuerza muscular en los tríceps y bíceps aumentando considerablemente hasta un 40%. Con estos resultados hacen la comparación de experimentos realizados por estadounidenses y rusos.

Dado que hoy en día, la electroestimulación se aplica con diferentes propósitos y generalmente implica inducir un potencial de acción en células excitables, nerviosas y/o musculares mediante un campo magnético. Es por eso que se deben tener presentes varios parámetros importantes, como lo son: el tipo de corriente, la intensidad, la cantidad de energía y los tiempos de aplicación.

Todos estos parámetros elementales en conjunto, deben dar como resultado una corriente eléctrica que sea capaz de crear una contracción con óptima calidad, sin riesgos de quemaduras por fricción descontrolada u otros inconvenientes que se vinculan con el uso de la electricidad. Esta determinación de los parámetros de la corriente eléctrica requiere forzosamente conocimientos de la física elemental.

2.2 Corriente eléctrica en electroterapia

La corriente eléctrica es la cantidad del flujo que recorren los electrones en un material o medio conductor durante un periodo de tiempo determinado, desde los extremos puntuales a considerar como referencia; presentándose así también una diferencia de potencial. La energía que se produce con el desplazamiento de las partículas electrónicas en un cable puede ser calorífica, lumínica, magnética o todas ellas. Generalmente los medios conductores se encuentran en las distintas formas en que se presenta la materia, como líquidos, sólidos y gases. El transporte más común de los electrones con fines industriales y domésticos son los materiales solidos por mencionar el cobre, ésta tiene una velocidad de desplazamiento menor a 1mm; entendiéndose que dicha velocidad varía con las características de cada medio conductor.

2.2.1 Aspectos fisiológicos de la corriente eléctrica

La contracción de forma voluntaria en el ser humano es realmente simple, el cerebro emite una orden llamado potencial de acción. La velocidad de propagación estándar se considera de 30m a lo largo del medio motor, donde se invierte la polaridad de las células que va atravesando. El recorrido se finaliza con la intervención del neurotransmisor acetilcolina. Dicho potencial de acción se manifiesta como dato de información hacia el interior de la célula muscular para obtener un resultado corto.

Por tanto la electroestimulación (EEM) tiene un objetivo muy similar para los músculos, también en el medio conductor de dicho nervio motor, pero en este caso con la aplicación de diferencias de potenciales o tensiones para desplazar el flujo de corrientes eléctricas externas y controladas.

La provocación artificial de contracciones musculares mediante una corriente interrumpida o una corriente alterna, puede perseguir diferentes objetivos: pueden ser denervaciones completas o parciales de fibras musculares, donde el interés se centrará en "normalizar" el músculo; o en caso contrario será hipertrofiarlo, cuando la inervación del músculo sea normal. En cualquier caso los objetivos más importantes que se buscan son: tonificación de la musculatura, mejora del riego sanguíneo, recuperación de la sensibilidad muscular, obtención de información de la excitabilidad eléctrica de fibras nerviosas y tejido muscular, retardo o impedimento de la atrofia muscular, prevención de la fibrosis muscular, estiramiento muscular e incluso la relajación del músculo.

Generalmente, un paciente experimenta una disminución o incluso la desaparición de la corriente conforme pasa el tiempo. Este proceso se conoce como "*acomodación*" o "*adaptación*" y se debe a que los receptores estimulados pasan información sobre los cambios externos en un grado menor. La estimulación con una señal invariable conduce a una disminución del efecto estimulante. Se puede evitar la *acomodación* de las siguientes maneras:

- Incrementando la amplitud
- Variando la frecuencia
- Realizando ajustes para tener una modulación baja [7].

2.2.1.1 Fenómenos no nocivos causados por la electricidad en el ser humano

Sensación eléctrica: es el valor mínimo de corriente que causa alguna sensación en la persona por la que fluye. El posicionamiento y tamaño de los electrodos, así como la frecuencia utilizada en la estimulación son factores fundamentales.

Límite de tolerancia (implica sensaciones intensas incluyendo el dolor): este depende de la densidad de la piel así como del tamaño de los electrodos. El límite de tolerancia varía de igual modo dependiendo del estado de salud o la condición patológica de la persona.

Dolor: la sensación de dolor puro no puede experimentarse en ningún punto de la piel. El dolor como tal es difícil de lograr, ya que para ello se necesita que existan únicamente receptores al dolor en el área estimulada.

Estimulación neuromuscular: aumentar la fuerza muscular, la circulación de la sangre y la

amplitud de movimientos, es posible disminuir los espasmos musculares.

Impotencia muscular

2.2.2 Tipos de Corriente eléctrica

2.2.2.1 Corriente eléctrica en estado constante

La corriente en estado constante es la corriente ininterrumpida con una sola dirección del flujo de los electrones y la intensidad constante en un medio conductor determinado. Esta corriente puede ser con polaridad positiva y negativa, variando la intensidad entre un mínimo que no es precisamente cero y el máximo, puede ser senoidal, cuadrada, triangular, impulsional o la mezcla de múltiples formas.

Cabe mencionar que el paso del flujo de electrones en el cuerpo no se lleva a cabo por el método de la conducción, como ocurre con los medios conductores, si por el desplazamiento de cargas eléctricas. El carácter unidireccional de la corriente continua le da la propiedad de desplazar iones libres en las soluciones salinas, lo que implica un trasporte activo de sustancias por convección. El método de transporte de convección implica que el trayecto de la intensidad de corriente eléctrica se hace produciendo variaciones y orientaciones iónicas, de las cuales se presentan modificaciones de la fisiología de las membranas celulares.

Muchas de las orientaciones iónicas desencadenan reacciones bioquímicas en diferentes sitios para dar como manifestación respuestas también fisiológicas o terapéuticas. La corriente galvánica en los estímulos actúa a distancia en los ejes *hipofisosuprarrenales* desatando así la liberación de la *endorfina hipofisaria*.

2.2.2.1.1 Galvanización

Es la corriente eléctrica continua en estado permanente que se mantienen con la misma polaridad durante un tiempo determinado, donde la intensidad aplicada aumenta progresivamente hasta alcanzar el nivel deseado, y después de un tiempo establecido disminuye su nivel progresivamente hasta cero. La corriente se aplica mediante dos electrodos metálicos con iguales

dimensiones, que se sitúan en dos regiones consideradas para el tratamiento. Con el electrodo de polaridad negativa se puede obtener la vasodilatación activa y el incremento de la excitación neuromuscular (catelectrotono). En tanto, para el electrodo con polaridad positiva tiene como resultado una acción sedativa y la disminución de la excitación neuromuscular (anelectrotono). Los resultados generalizados son las reacciones vasomotoras, y para que uno de los dos efectos predomine en la zona o área a tratar se coloca un electrodo activo con superficie más pequeña que el otro. El comportamiento del paso de la corriente galvánica entre los electrodos separados con un ángulo de 180° se presenta también con una disminución de la densidad de dicha corriente entre la piel y la profundidad del músculo. Cuando la colocación de ellas es de 90°, se obtiene un aumento en la zona. En cambio a 50°, resulta una máxima densidad de la corriente eléctrica, donde se acentúa más en las cercanías de la piel que en el hueso, como se muestra en la Figura 2.1.

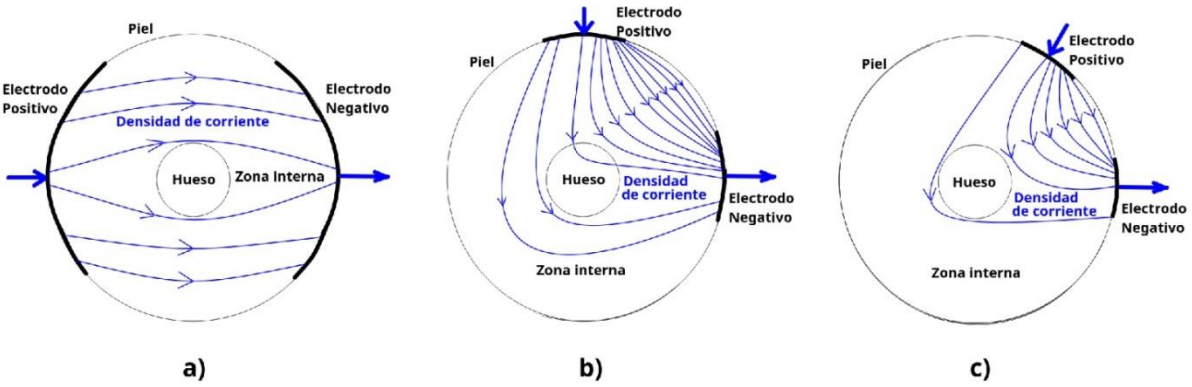

Figura 2.1: Densidad del flujo de la corriente eléctrica con a) electrodos anchos y ángulo de 180° generan disminución, b) electrodos a 90° generan aumento y c) electrodos en 50° se presenta la máxima densidad.

2.2.2.1.2 Ionización medicamentosa

También conocida como iontoforesis, que tiene como resultado la corriente galvánica para ionizar las sustancias en solución entre el electrodo y la piel, es decir, la propiedad de los iones o moléculas que penetran a través de los tegumentos. Hay soluciones que pueden contener cargas eléctricas en libertad, si se presenta una corriente unidireccional en la solución, del electrodo y la piel, ésta se transforma en un conductor, en tanto; el movimiento de los iones hace posible el paso de la corriente eléctrica. Se dice entonces que el ion positivo (del cual carece de un electrón) es atraído por el electrodo negativo, considerado también como cátodo, donde se presenta el

exceso de electrones. Por consiguiente, el ion negativo (que tiene un electrón suplementario) es desplazado hacia el electrodo positivo, identificado como ánodo carente del mismo [4].

Si se pretende introducir un radical medicamentoso en el organismo, se necesita conocer las propiedades electroquímicas, entonces se puede aplicar alguna fórmula que nos permita calcular (lo más aproximadamente posible) la cantidad de iones introducidos. Esta fórmula corresponde a la *Ley de Faraday*. Esta Ley establece que la cantidad en miligramos de una sustancia transportada por la corriente galvánica dependen de:

$$S_{mg} = \left[\frac{(P_{molecular})(I)}{(V_{ion})(K_{Faraday})}\right] t \tag{2.1}$$

donde:

S_{mg} , Dosis medicamentosa de la sustancia en miligramos

$P_{molecular}$, es el peso molecular

I, es la intensidad de corriente eléctrica en mA (miliamperes)

V_{ion}, es la valencia del ion o molécula

$K_{Faraday}$, es la constante de Faraday

t, es el tiempo en segundos

También se ha considerado un umbral de quemadura para la corriente galvánica que es de 0.1 mA/cm^2; y los tiempos de aplicación superiores a los 10*min* no proporcionaron mayor penetración a nivel subcutáneo. En la práctica el rango de la corriente galvánica está considerada con los valores estimados de 1 mA − 50 mA , y el tiempo de tratamiento con un lapso de 30*min* con sesiones diarias o alternas. Sin embargo, existe una tabla de valores que establecen unas directrices sobre la intensidad máxima según el área de tratamiento y el tamaño o dimensiones de los electrodos, como lo muestra la tabla 2.1.

Tabla 2.1: Intensidades de la *iontoforesis* recomendadas con base a la superficie corporal.

Electrodos	**Dimensión en cm**	**Área en cm²**	**Intensidad eléctrica mínimo y máximo**
Pequeños	3x4, 3x5	12, 15	0.6mA− 0.8mA, 0.75mA− 1mA
Medianos	6x8	48	2.4mA− 3.1mA

Grandes	8x12.5, 10x15	100, 150	5mA− 6mA, 7.5mA− 10mA

2.2.2.1.3 Parámetros de la corriente constante

a) Intensidad de corriente eléctrica

La intensidad de la corriente debe ser incrementada progresivamente tanto en la galvanización como en la ionización. Aunque la tolerancia del paciente es lo que regula la intensidad, ésta debe ser suficientemente elevada porque es directamente proporcional a la cantidad de electricidad y a la densidad de la corriente. Se debe tener en cuenta también que la intensidad de la corriente también aumenta proporcionalmente cuando la resistividad de la piel disminuye y su permeabilidad aumenta. La intensidad de 0,1 mA/cm^2 de la superficie del electrodo activo representa un valor límite (umbral de quemadura). En general, la intensidad utilizada y bien tolerada es de 0,05 mA/cm^2.

b) Resistencia e impedancia de la piel

Varía considerablemente según las personas y la región corporal. La piel callosa y seca de un trabajador manual puede tener una resistencia de 40,000 a 100,000 Ω, mientras que la piel suave de una persona sedentaria tiene una resistencia que no supera los 5,000 Ω.

c) Conductividad eléctrica

Depende del contenido de agua y electrólitos de los tejidos. Por ejemplo, el músculo y el cerebro muestran mejor conductividad, mientras que los tendones y las fascias tienen gran resistividad.

d) Densidad de la corriente

Es el factor más importante de la acción local. La densidad de una corriente es proporcional a la intensidad de la misma e inversamente proporcional a la superficie del electrodo. Cuanto más elevada sea la densidad de la corriente, mayor será la penetración.

e) pH

Puede actuar sobre la polaridad de la sustancia medicamentosa ionizada. En general, la mayoría de las soluciones tienen un pH cercano a 4. Un incremento del pH no es compatible con la utilización terapéutica humana, ya que provocaría una quemadura grave [4].

2.2.2.2 Corriente eléctrica en estado variable

En este tipo de corriente, la frecuencia e intensidad pueden variar en el tiempo, proporcionando impulsos cuya forma frecuencia y duración determinan algunas características biológicas. Los impulsos aislados no constituyen una corriente, pero si se aplican de manera reiterada y en forma de trenes de onda sin interrupción si generan corriente. Esta clasificación generalmente hace referencia a familias de frecuencias: baja (menores a 1,000 Hz), media (entre 1,000 y 10,000 Hz) y alta (mayores de 10,000 Hz)[4].

2.2.2.2.1 Tipos de corrientes variables

Los tipos de corrientes utilizados en la actualidad en electroterapia se muestran en la Figura 2.2.

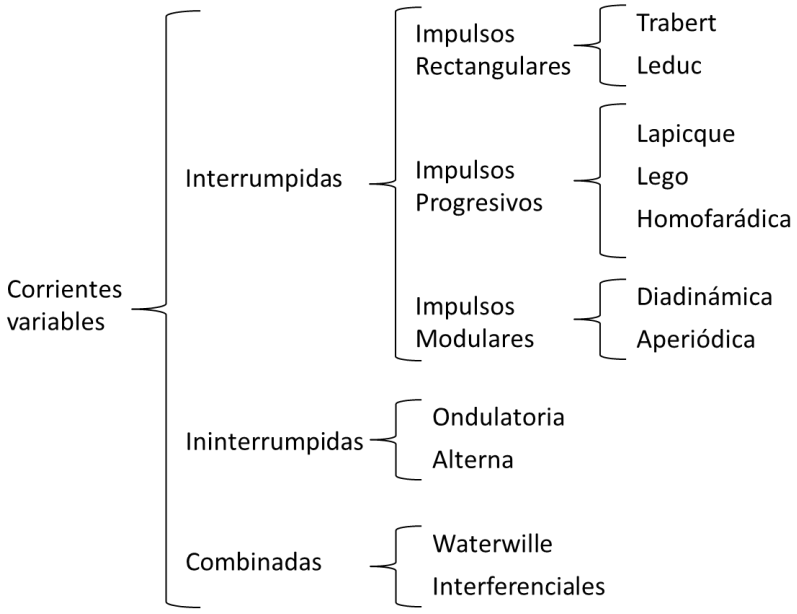

Figura 2.2: Cuadro sinóptico de la clasificación de corrientes variables según su forma, utilizadas en electroterapia [8].

2.2.2.2.2 Parámetros de la corriente variable

a) Forma de los impulsos

Los impulsos utilizados pueden generarse de distintas formas siendo las más comunes: rectangular, de pendiente positiva, exponencial, homofarádico (onda breve y puntiaguda), bifásico y senoidal, como se muestran en la Figura 2.3.

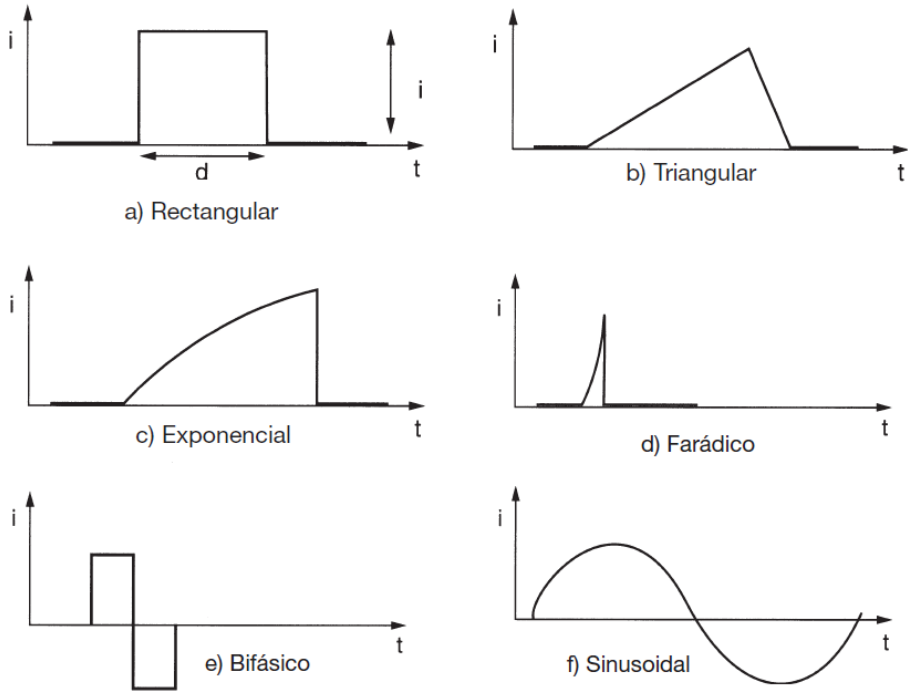

Figura 2.3: Forma de impulsos aislados utilizados en electroterapia [4].

b) Modulación

Es el paso de uno a otro modo de corriente cambiando uno o más parámetros de variación, circunstancia que puede modificar la naturaleza y las propiedades biológicas de la corriente inicial.

c) Trenes de impulsos no modulados

Las diferentes formas de corriente utilizadas en terapias se caracterizan por la forma de los impulsos, la frecuencia, la intensidad y el modo de producción, que a su vez les confieren sus propiedades terapéuticas. Algunos ejemplos de estos trenes se muestran en la Figura 2.4.

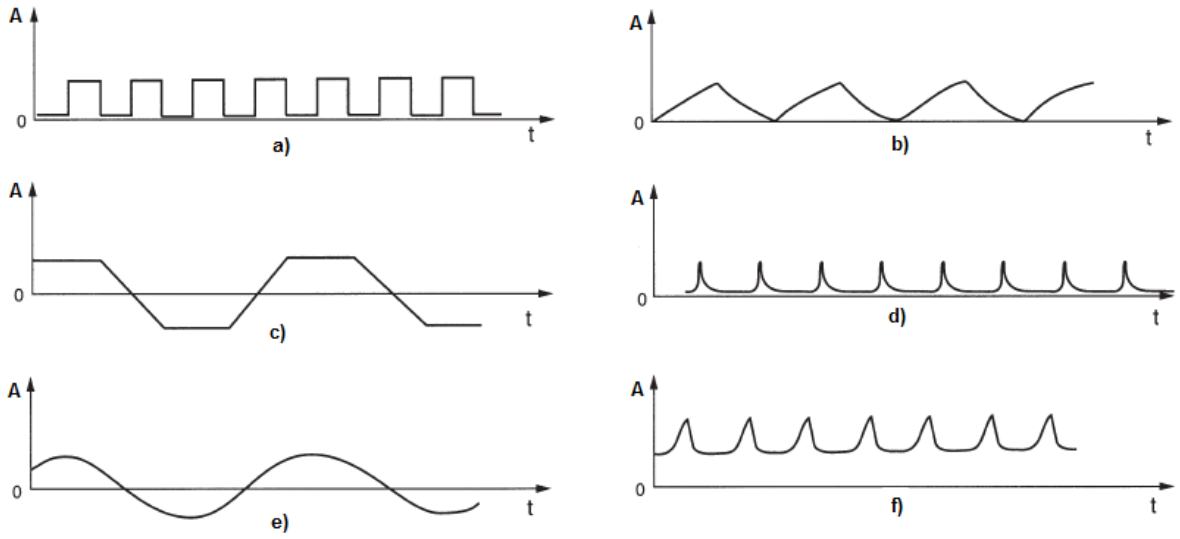

Figura 2.4: Trenes de impulsos no modulados. Dependiendo de la forma, estos trenes reciben un nombre como pueden ser: a) corriente de Leduc; b) corriente de Lapicque; c) corriente de Laquérière; d) corriente farádica; e) corrientes sinusoidales; f) corriente galvanofarádica.

2.2.2.2.3 Corrientes Interferenciales

Este tipo de corrientes son el resultado de la mezcla de dos corrientes alternas de media frecuencia, de ahí su nombre de corrientes interferenciales. El desarrollo del proyecto se centra en la generación de este tipo de corrientes, por lo que a continuación de describe más a detalle sus características.

Las corrientes interferenciales son utilizadas en medicina a partir de las experiencias llevadas a cabo por el Dr. Nemec, médico europeo que investigó la acción de las corrientes eléctricas sobre el cuerpo humano. Estas tuvieron gran aceptación en países como Francia y Alemania, pero fue en la década de los años 60's y 70's donde esta modalidad de electroterapia alcanzó un nivel de mucha importancia, haciéndose conocer en casi todo el mundo. La experiencia del Dr. Nemec sobrevino por necesidad de conseguir un tipo de onda eléctrica que consiguiera acceder a profundidades mayores que con otros tipos de corrientes estimulantes, así como permitir la aplicación de intensidades altas sin ocasionar molestias excesivas [7].

El proceso de generación se basa en dos corrientes de frecuencia media que se encuentran en la coexistencia en tiempo y espacio, pero con una diferencia de 1 a 150 Hz entre sí, Figura 2.5. Las señales se dirigen hacia una parte del cuerpo a través de los circuitos electrónicos

generadores, cada uno de los cuales tiene dos electrodos. Ambas corrientes están dispuestas de tal manera interfieren entre sí, dando lugar a la creación de una nueva corriente eléctrica resultante de la diferencia entre ambas, la corriente interferencial; así se consigue obtener frecuencias de estimulación variables desde 1 a 150 Hz. Para garantizar una excitación activa en todas las estructuras tisulares de la región de superposición, es posible adicionar un sistema que permita generar una corriente de interferencia dinámica, aumentando la intensidad de un circuito y disminuyendo la del otro, de esta forma, la intensidad total se mantiene y el efecto de rotación de la interferencia integral se realza.

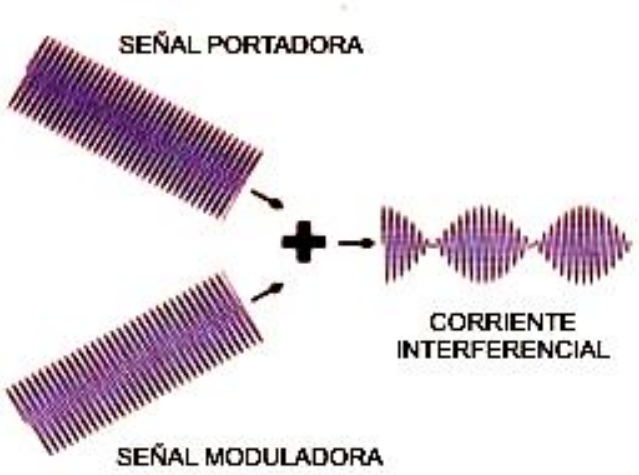

Figura 2.5: Curva envolvente de frecuencia media. La señal resultante es una curva con un componente de baja frecuencia.

2.3 Modelos matemáticos para generación de barridos.

2.3.1 Modelo matemático de la función trapezoidal.

Para obtener la función trapezoidal por métodos convencionales, primero deben identificarse las expresiones que involucren la magnitud o altura con respecto al tiempo, como lo muestra la siguiente Figura 2.6:

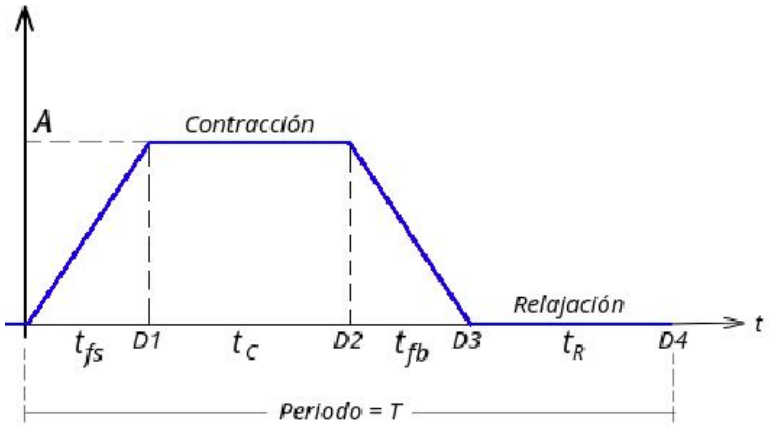

Figura 2.6: Forma del trapecio y la altura máxima con los tiempos como parámetros de control.

Flanco de subida Fs, se considera la fórmula de la pendiente $\mathrm{m} = \frac{\mathrm{Y_2 - Y_1}}{\mathrm{X_2 - X_1}}$:

$$m_S = \frac{A}{D_1}; Y_2 - Y_1 = m_S(X_2 - X_1), Y_2 - 0 = \frac{A}{D_1}(X - 0)$$

$$f_{FS}(t) = \frac{A}{D_1}t; t_{fs} = 0 \rightarrow D_1 \tag{2.2}$$

En tanto t_c, que es el tiempo de contracción, si permanece constante entonces:

$$f_{AC}(t) = A; t_C = D_1 \rightarrow D_2 \tag{2.3}$$

Ahora, en el tiempo de flanco de bajada t_{fb}, la pendiente está dada por:

$$m_b = -\frac{A}{D_3 - D_2}; Y_{FB}(t) - A = -\frac{A}{D_3 - D_2}(t - D_3)$$

$$f_{FB}(t) = A\left[\frac{D_3 - t}{D_3 - D_2}\right]; t_{fb} = D_2 \rightarrow D_3 \tag{2.4}$$

Entonces, en el tiempo de relajación t_R, su comportamiento es:

$$f_{AR}(t) = 0; t_R = D_3 \rightarrow D_4 \tag{2.5}$$

Por lo anterior, la función *f(t)* considerando las ecuaciones 2.2, 2.3, 2.4 y 2.5, se define como:

$$f_{Trap}(t) = \begin{cases} \frac{A}{D_1}t, & \text{si} \quad 0 \le t \le D_1 \\ A, & \text{si} \quad D_1 \le t \le D_2 \\ A\left[\frac{D_3 - t}{D_3 - D_2}\right], & \text{si} \quad D_2 \le t \le D_3 \\ 0, & \text{si} \quad D_3 \le t \le D_4 \end{cases} \tag{2.6}$$

2.3.2 Modelo matemático de la función cuadrada

Para obtener la función de la onda cuadrada, se lleva a cabo un análisis de la misma forma como en la función trapezoidal, como se muestra en la Figura 2.7:

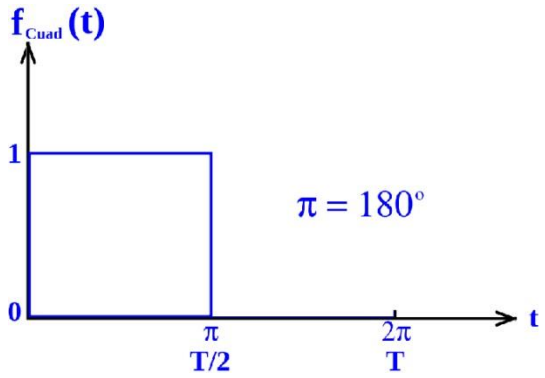

Figura 2.7: Forma de onda cuadrada y las características que la conforman como señal inferencial.

$$f_{Cuad}(t) = \begin{cases} 1, & \text{si} \quad 0 \leq t \leq \pi \\ 0, & \text{si} \quad \pi \leq t \leq 2\pi \end{cases} \tag{2.7}$$

2.3.3 Serie de Fourier trigonométrica

2.3.3.1 Función trapezoidal para barridos.

Sea la función $f(t)$ una función periódica con periodo T, la cual se puede representar por la serie trigonométrica:

$$f(t) = \frac{1}{2}a_o + \sum_{n=1}^{\infty}\left[a_n Cos(n\omega_0 t) + b_n Sen(n\omega_0 t)\right] \tag{2.8}$$

dónde: $a_0, a_n \; y \; b_n$, son los coeficientes de Fourier denotados como:

$$a_o = \frac{2}{T}\int_{-T/2}^{+T/2} f(t)dt \tag{2.9}$$

$$a_n = \frac{2}{T}\int_{-T/2}^{+T/2} f(t)Cos(n\omega_o t)dt, n = 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,\ldots,+\infty \tag{2.10}$$

$$b_n = \frac{2}{T}\int_{-T/2}^{+T/2} f(t)Sen(n\omega_o t)dt, n = 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,\ldots,+\infty \tag{2.11}$$

ω_0, es la función angular que define:

$$\omega_o = 2\pi f = \frac{2\pi}{T};\ f = \frac{1}{T} = frecuencia = [Hertz = Hz];\ T = Periodo = [Segundos = s] \tag{2.12}$$

El coeficiente a_0, se obtiene de la siguiente forma:

$$f(t) = \frac{A}{D_1}t + A + \frac{AD_3}{(D_3 - D_2)} - \frac{A}{(D_3 - D_2)}t + 0$$

$$a_o = \frac{2}{T}\left[\int_o^{D_1} \frac{A}{D_1}t\,dt + \int_{D_1}^{D_2} A\,dt + \int_{D_2}^{D_3} A\frac{D_3}{D_3 - D_2}dt - \int_{D_2}^{D_3} \frac{A}{D_3 - D_2}t\,dt + \int_{D_3}^{D_4} 0\,dt\right]$$

$$a_o = \frac{A(D_3 + D_2 - D_1)}{T} \tag{2.13}$$

El coeficiente a_n, se obtiene:

$$a_n = \frac{2}{T}\left[\frac{A}{D_1}\int_0^{D_1} t\,Cos(n\omega_o t)dt + A\int_{D_1}^{D_2} Cos(n\omega_o t)dt + \frac{AD_3}{(D_3 - D_2)}\int_{D_2}^{D_3} Cos(n\omega_o t)dt \right.$$
$$\left. - \frac{A}{(D_3 - D_2)}\int_{D_2}^{D_3} t\,Cos(n\omega_o t)dt + 0\int_{D_3}^{D_4} Cos(n\omega_o t)dt\right]$$

$$a_n = \frac{2A}{T(n\omega_o)^2}\left[\frac{(D_3 - D_2)[Cos(n\omega_o D_1) - 1] + D_1[Cos(n\omega_o D_2) - Cos(n\omega_o D_3)]}{D_1(D_3 - D_2)}\right] \tag{2.14}$$

De la misma manera, el coeficiente b_n, se obtiene como:

$$b_n = \frac{2}{T}\left[\frac{A}{D_1}\int_0^{D_1} t\,Sen(n\omega_o t)dt + A\int_{D_1}^{D_2} Sen(n\omega_o t)dt + \frac{AD_3}{(D_3 - D_2)}\int_{D_2}^{D_3} Sen(n\omega_o t)dt \right.$$
$$\left. - \frac{A}{(D_3 - D_2)}\int_{D_2}^{D_3} t\,Sen(n\omega_o t)dt + 0\int_{D_3}^{D_4} Sen(n\omega_o t)dt\right]$$

$$b_n = \frac{2A}{T(n\omega_o)^2}\left[\frac{(D_3 - D_2)[Sen(n\omega_o D_1)] + D_1[Sen(n\omega_o D_2) - Sen(n\omega_o D_3)]}{D_1(D_3 - D_2)}\right] \tag{2.15}$$

Finalmente, la expresión completa de la serie de Fourier Trigonométrica está dada por tres funciones que se escriben separadas, aunque todas ellas conforman una sola, esto es:

$$f_{Trapezoidal}(t) = \frac{A(D_3 + D_2 - D_1)}{2T} +$$

$$\sum_{n=1}^{\infty}\left[\frac{2A}{T(n\omega_o)^2}\left[\frac{(D_3 - D_2)[Cos(n\omega_o D_1) - 1] + D_1[Cos(n\omega_o D_2) - Cos(n\omega_o D_3)]}{D_1(D_3 - D_2)}\right]Cos(n\omega_o t)\right] +$$

$$\sum_{n=1}^{\infty}\left[\frac{2A}{T(n\omega_o)^2}\left[\frac{(D_3 - D_2)[Sen(n\omega_o D_1)] + D_1[Sen(n\omega_o D_2) - Sen(n\omega_o D_3)]}{D_1(D_3 - D_2)}\right]Sen(n\omega_o t)\right] \quad (2.16)$$

Nota: recordemos que $n = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, ..., +\infty$ para todo $f(t)$.

2.3.1.2 Función cuadrada para señales interferenciales.

En el caso de la función cuadrada, está es definida de la forma:

$$f_{Cuadrada}(t) = \frac{1}{2} + \frac{2}{\pi}\sum_{n=1}^{\infty}\left[\frac{Sen[(2n-1)\ \omega_{Cuad}\ t_{Cuad}]}{2n-1}\right] \quad (2.17)$$

donde: $f_{Cuadrada}(t)$, función de la onda cuadrada

ω_{Cuad}, frecuencia angular de barridos de la función cuadrada en rad/s

t_{Cuad}, tiempo de barridos de la función cuadrada en segundos

n = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, …, +∞ para toda $f_{Cuadrada}(t)$

2.4 Onda Rusa portadora de señal

El Dr. Yakov Kotz, médico ruso, en una conferencia de 1977, dio a conocer el descubrimiento de una onda eléctrica muy particular con el cual realizó experimentos y como resultado, obtuvo el incremento de fuerza muscular en atletas rusos de hasta un 40%. Los tratamientos originalmente fueron considerados con frecuencias bajas y altas, con el objetivo de encontrar una zona en particular del espectro electromagnético para ser usadas con fines óptimos. En tanto también se dieron resultados con base a experimentos con humanos en diferentes días y modos de electroestimulación muscular, en la mayoría de los casos de las extremidades (brazos y piernas).

Sin embargo, el Dr. Kotz no profundizó en una modulación particular para acelerar resultados, pero sí de mantener sin deformidad la onda con amplitud y frecuencia constante. Con el tiempo los parámetros se fueron identificando y estandarizando, cada vez se consideraron otras formas de onda para la terapia física, al grado de diseñar equipos más complejos y corrientes interferenciales completamente moduladas, para cada tratamiento.

Los equipos que generan Ondas Rusas se diseñan para lograr una estimulación de los nervios motores, con la mayor penetración o profundidad posible, y con la menor sensación dolorosa para el paciente. Las Ondas Rusas pueden producir tensiones en los músculos mayores que las voluntarias, estas su utilizan para lograr:

- Contracción de músculos denervados parcialmente.
- Reducción de edemas.
- Reducción del dolor.
- Aumento de la circulación.
- Aumento de la velocidad de contracción.
- Aumento del volumen muscular.

La forma de la Onda Rusa es como se muestra en la Figura 2.8, donde se puede observar que la onda comienza de cero, aumenta la amplitud durante un tiempo al cual lo llamamos RAMPA DE SUBIDA; posteriormente mantiene esta amplitud sin variar durante un tiempo que se llama CONTRACCIÓN; después decrece su amplitud durante un tiempo al cual lo llamamos RAMPA DE BAJADA y finalmente se mantiene en cero para producir un descanso durante un tiempo que se llama RELAJACIÓN. Todos estos parámetros son regulados en forma digital, o sea, para efectuar una aplicación se deberán poner los valores correspondientes para el tipo de terapia que se desee [5].

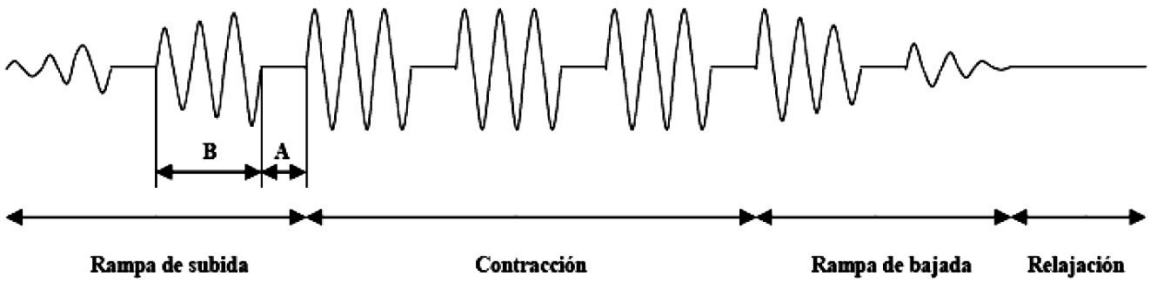

Figura 2.8: Forma de onda rusa y sus diferentes fases

2.4.1 Modulación en Amplitud.

La señal portadora en efecto es la *Onda Rusa*, que consiste en una función senoidal simétrica para CA con una frecuencia de 2.5kHz, amplitud unitaria y con un off-set (desplazamiento en nivel de DC) para el procesamiento digital como lo muestra la Figura 2.9, los parámetros considerados son:

Amplitud : 1

Muestras : 50

Cuantización: 0-255

f_{Rusa} : 2,500Hz

P_{Rusa} : 0.4ms

t_{MRusa} : 8ms = 0.008ms

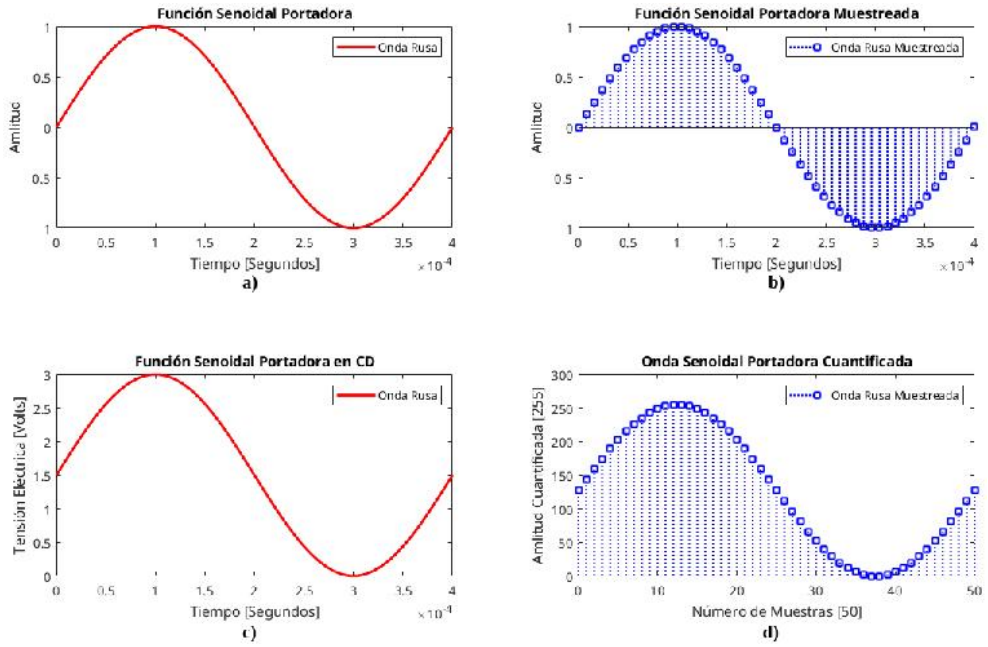

Figura 2.9: a) Señal base con amplitud unitaria y periodo de 4ms. b) Señal muestreada a razón de 50 muestras por ciclo. c) Señal base con off-set. d) Onda rusa final cuantizada de 0 a 255 niveles.

El proceso de modulación consiste en obtener crecimientos y decrecimientos de una onda senoidal con frecuencia constante de 2,500Hz; mediante una función trapezoidal, que se caracterice por controlar los barridos y que comprenda de regiones de contracción y relajamiento, con rampas de subida y bajada, Figura 2.10. De tal manera que también sea gradual la forma de la

señal interferencial, y que permita modificar dichas rampas si es que se desea para tratamientos específicos. Esto se logra mediante el cálculo en software con la siguiente expresión:

$$F_{Kotz}(t) = \left[F_{Trapezoidal}(t)\right][Sen(t)] \tag{2.18}$$

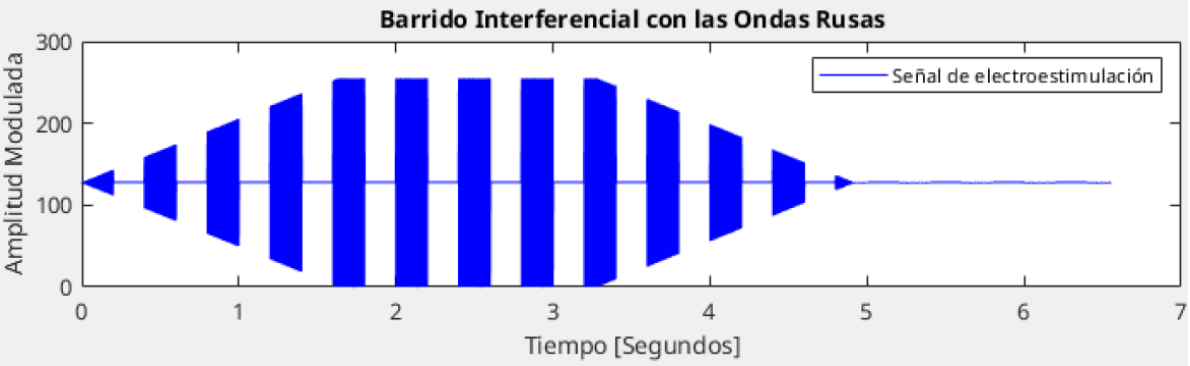

Figura 2.10. Onda Kotz modulada con una función trapezoidal, utilizada en tratamientos de analgesia.

Capítulo 3

Tecnologías para Diseño de Sistemas Empotrados.

En este capítulo se hace una descripción general de las diversas tecnologías existentes con las que se pueden desarrollar sistemas de aplicación específica, en este caso un generador de ondas rusas. Se realiza un comparativo entre las tecnologías disponibles describiendo sus ventajas y desventajas. Finalmente se hace la justificación de la tecnología elegida para el desarrollo del sistema.

3.1 Antecedentes.

Los sistemas empotrados (SE) son normalmente definidos como el software implementado en hardware a manera de realizar funciones específicas en tiempo real, por lo que generalmente se integran como parte de un sistema más grande. Hoy en día, los SE se encuentran en todas partes, desde productos eléctricos y electrodomésticos, hasta mecanismo de compensación no lineal, sistemas de automatización complejos y sistemas de control adaptativo. En comparación con la plataforma informática (computadoras), los SE normalmente tienen mucho menos potencia de cómputo y tamaño de memoria muy limitado. Sin embargo, para resolver tareas particulares en tiempo real, las desventajas anteriores a su vez se convierten en virtudes, ya que los SE son menos costosos y mucho más fáciles de diseñar. El diseño más simple se refleja en ambos hardware y software. Diseño fijo o solo variaciones limitadas del hardware permite el uso de sistemas operativos (SO) simplificados que permiten una operación predecible en tiempo real o incluso una implementación directa de aplicaciones sin ningún sistema operativo formal [9]. Diversas tecnologías digitales son utilizadas para este fin: microprocesadores (Microprocessor Unit, MPU), microcontroladores (Microcontroller Unit, MCU), procesadores digitales de señal (Digital Signal Processor, DSP), matriz de compuerta programable en campo (Field Programmable Gate Array, FPGA) y tecnologías de circuito integrado de aplicación específica (Application Specific Integrated Circuit, ASIC). Estas tecnologías generalmente son evaluadas acorde a la aplicación, con la finalidad de seleccionar aquella que sea la más rentable y proporcione el más alto rendimiento.

3.2 Factores a considerar en el desarrollo de aplicaciones.

El desarrollo de un sistema utilizando tecnología digital busca minimizar, simultáneamente, cuatro factores:

1. El número de transistores empleados, lo que afectará el tamaño físico del sistema, el costo unitario y el consumo de energía. Los avances en la tecnología de proceso han reducido continuamente a lo largo de los años el área del transistor, pero el consumo de energía estático y dinámico depende del número de transistores. Por lo tanto, la cantidad de transistores utilizados sigue siendo una métrica importante de la eficiencia del sistema.

2. El número de ciclos de reloj necesarios, lo que afecta el rendimiento y el consumo de energía. El incremento en las frecuencias de reloj de los dispositivos asociadas con geometrías de proceso más pequeñas, han permitido más ciclos de reloj en un intervalo de tiempo dado, pero a expensas de un mayor consumo de energía. Menos ciclos de reloj a menudo significa menos consumo de energía.

3. El tiempo necesario para el desarrollo de la aplicación, lo cual impacta fuertemente en su aceptación en el mercado. Un producto que pierde su ventana de mercado es un desperdicio total del esfuerzo realizado. Es importante considerar que en muchos casos, el desarrollo de software lleva más tiempo y cuesta más que el del hardware.

4. Costos de ingeniería no recurrentes como la fabricación de prototipos y el costo del desarrollo de hardware y software. El aumento de estos costos asociados con las tecnologías de punta necesarias para el proceso, puede dejar a muchas aplicaciones fuera del alcance para su realización [10].

La elección de la tecnología para el desarrollo de cualquier aplicación es un compromiso de ingeniería. En la mayoría de los casos, esta elección depende de una combinación compleja de factores y generalmente ninguna tecnología digital de forma independiente cubre todas las necesidades. Las diferentes mezclas tecnológicas suelen ser más apropiadas en las diferentes etapas del ciclo de vida del producto final.

3.3 Tecnologías digitales para diseño de sistemas empotrados.

3.3.1 Microprocesador (MPU):

Una unidad de microprocesador independiente (MPU) agrupa una CPU (Central Processing Unit) con interfaces periféricas como memoria (como DDR3 o DDR4), buses (como USB), interfaces de comunicación Ethernet, Bluetooth y más, Figura 3.1, por lo que estos sistemas son flexibles y versátiles y están diseñados para trabajar con diversos sistemas operativos de alto nivel como lo son: Windows, Linux, Unix, etc., [11].

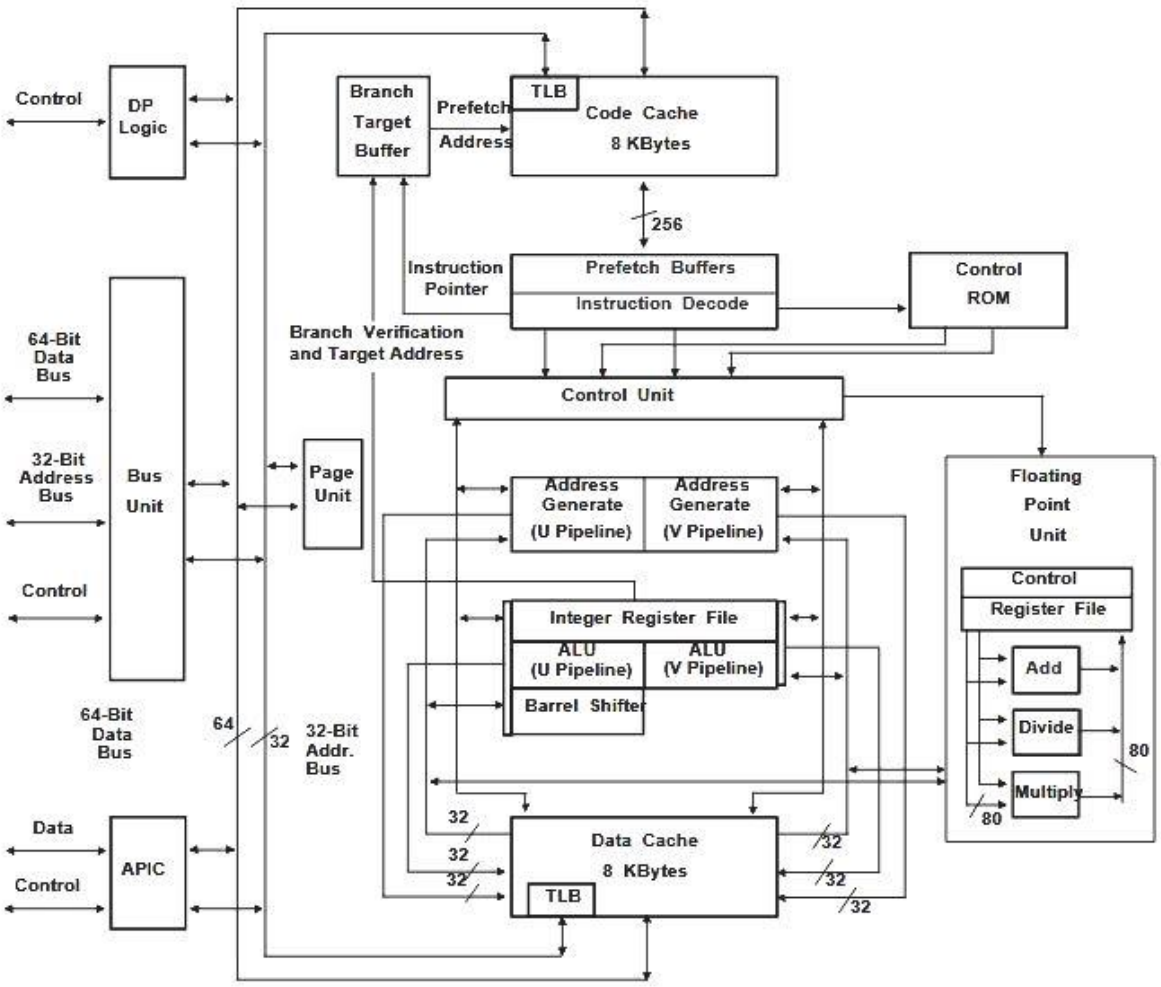

Figura 3.1. Diagrama a bloques de un microprocesador Intel Pentium [12].

Los MPU tienen gran influencia en las computadora (portátil y de escritorio). La arquitectura de hardware de la CPU de estos dispositivos está diseñada para uso general, pero a menudo incluye bloques especializados, como lo es la unidad de punto flotante (Floating Point Unit, FPU) para operaciones matemáticas. Las CPU de gama baja ejecutan operaciones de manera secuencial, pero núcleos de multiprocesamiento son cada vez más estándar en los en los sistemas de gama alta. Por ejemplo el Intel-Xeon Gold 6148 tiene hasta 20 núcleos [13].

3.3.2 Microcontrolador (MCU):

Una unidad de microcontrolador (MCU) combina el núcleo de la CPU con la memoria interna y algunos periféricos digitales y analógicos integrados en un chip, Figura 3.2 , donde normalmente se ejecuta un sistema operativo simplificado. A manera de reducir los costos de

manufactura y potencia de operación, algunos MCU están diseñados para trabajar con una muy corta longitud de palabra, por ejemplo de hasta 4 bits o en el otro extremo de hasta 64 bits; tienen poca RAM (Random-Access Memory); operan con reloj en rangos desde kHz (kilo Hertz) hasta algunos MHz (Mega Hertz); y algunos mantienen parcialmente su funcionalidad con parte de su circuitería en suspensión hasta que un evento de interrupción es registrado. Los microcontroladores se encuentran en muchas partes de las aplicaciones integradas como: paneles de instrumentos automotrices, controladores de motores, tarjetas inteligentes, etc.

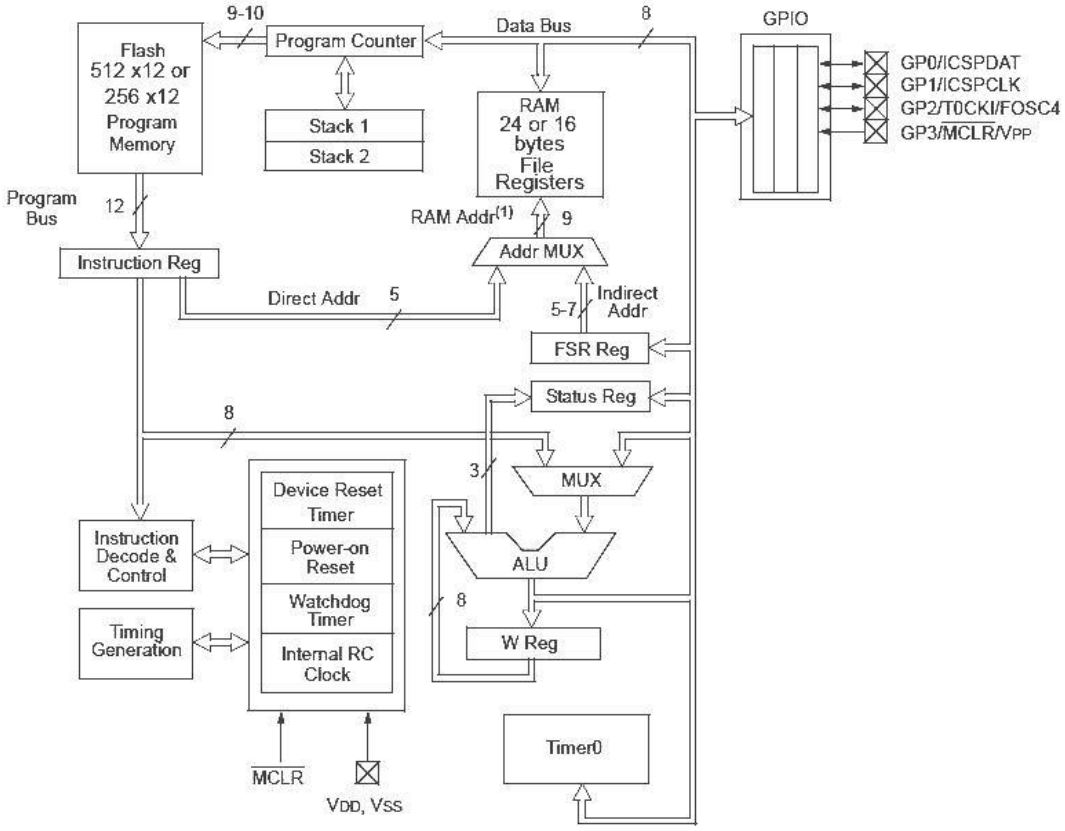

Figura 3.2: Diagrama a bloques de un microcontrolador PIC10F200 de Microchip [14]. Bus de datos de 8 bits, 4 terminales para comunicación con el exterior.

3.3.3 Procesador Digital de Señales (DSP)

Los procesadores de señal digital (DSP) están diseñados para tener integrados multiplicadores y bloques DSP, lo que les permite realizar operaciones aritméticas complejas, las cuales son fáciles de implementación con programación de alto nivel (por ejemplo en lenguaje C). Una de las principales virtudes del DSP es la capacidad de en un solo ciclo de reloj efectuar la operación multiplicación y acumulación. Además, los DSP tienen capacidades de procesamiento paralelo y bloques de memoria integrados, que mejoran en gran medida la

velocidad de procesamiento. Algunas arquitecturas DSP, llamadas controladores digitales de señal (DSC) están optimizados para aplicaciones de control y contienen periféricos orientados a esta tarea como: generadores PWM, temporizadores de vigilancia e interrupciones de respuesta rápida, Figura 3.3. Sin embargo, los DSP suelen tener un costo mucho mayor en comparación con otras tecnologías.

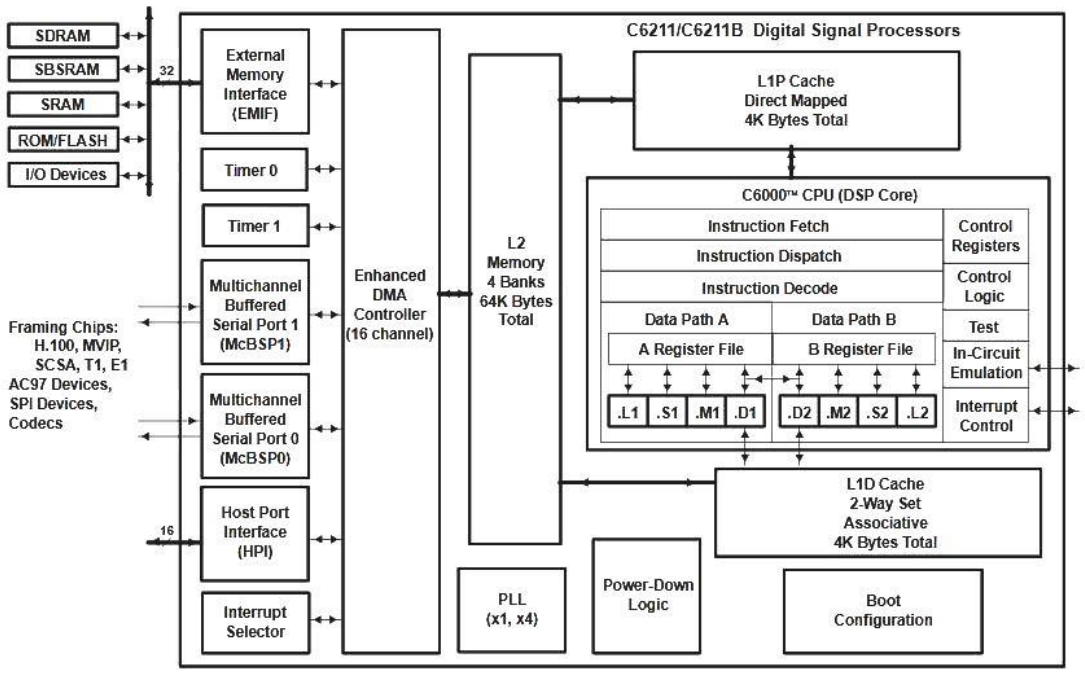

Figura 3.3. Diagrama a bloques de un DSP TMSC3206211 de Texas Instruments. Bus de datos de 32 bits. [15].

Generalmente, los DSP se aplican para el procesamiento de señales de audio e imagen, en aquellas donde el uso de un microcontrolador no es posible debido a las limitaciones computacionales. Una de las principales aplicaciones en la industria en el pasado fue como controlador de motor [9]. Hoy en día derivado del rápido crecimiento de los sistemas basados en la nube (internet), diversas aplicaciones como asistentes virtuales tales como: Google Voice, Alexa Voice Service con Amazon Web Services e IBM Watson, utilizan esta tecnología para reconocimiento e interpretación del contexto, con capacidad para almacenamiento de cantidades masivas de voz para acelerar el aprender y actualización algoritmos obteniendo una mayor exactitud y robustez [16].

3.3.4. Dispositivos reconfigurables (FPGA)

La tecnología FPGA fue desarrollada para sistemas empotrados digitales y se basa en la idea de tener matrices de bloques lógicos personalizados (Configurable Logic Block, CLB para el fabricante Xilinx; Logical Elements, LE para el fabricante Intel-FPGA) [17] rodeados por interconexiones de Entrada/Salida (Input-Output Bus, IOB) programables por el usuario, en los cuales se pueden personalizar las operaciones para una aplicación específica. A diferencia de los otros dispositivos mencionados, las conexiones entre bloques se pueden reprogramar fácilmente, cambiando la operación interna del hardware y permitiendo que un FPGA acomode los cambios en un diseño, o incluso que admita una nueva aplicación, durante la vida útil del dispositivo. La Figura 3.4 muestra la arquitectura conceptual de un FPGA.

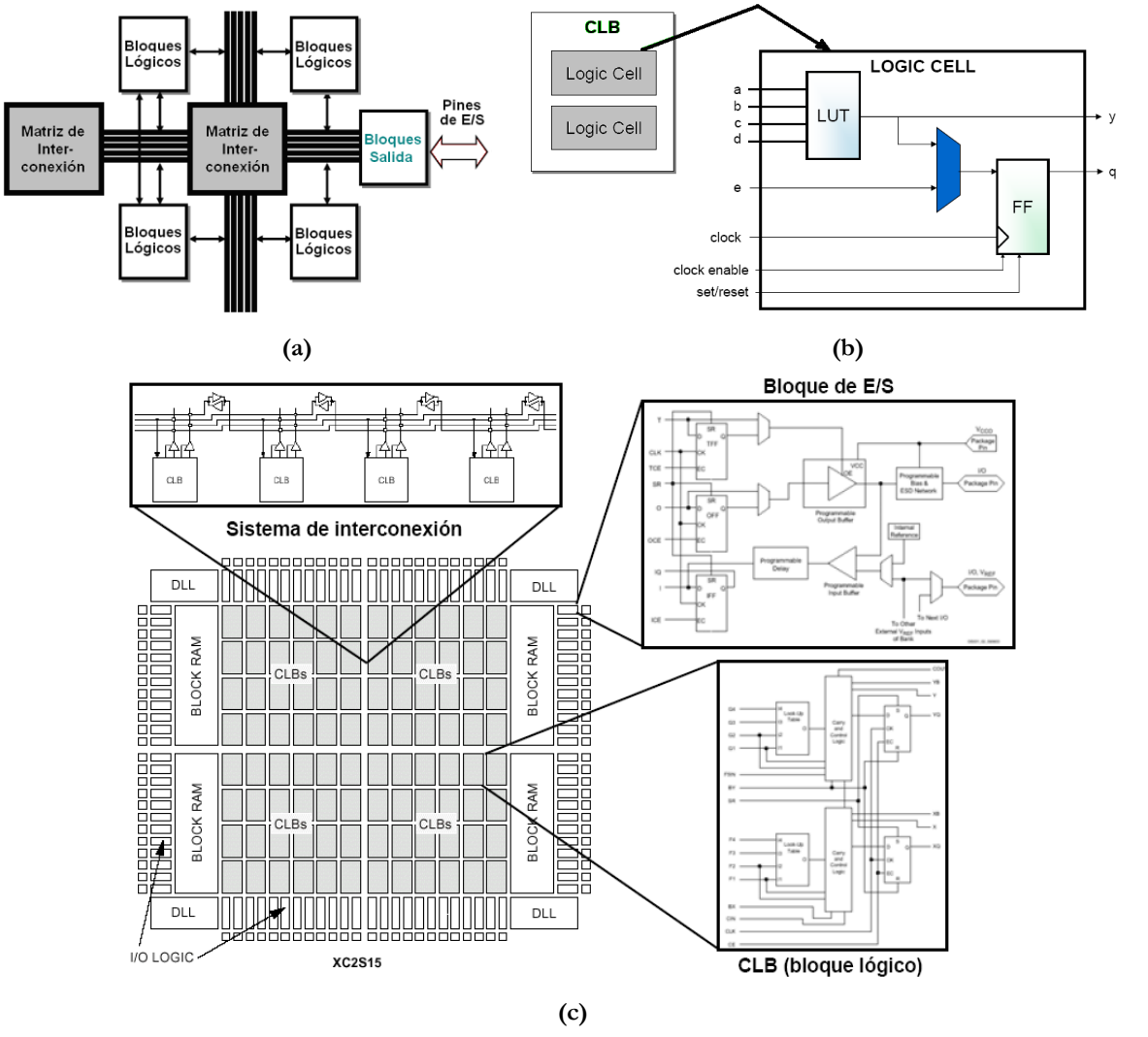

Figura 3.4 (a) Elementos de la lógica reconfigurable son: bloques lógicos, un sistema de interconexiones programables, bloques para entrada/salida y elementos como osciladores y divisores de reloj [17].

En los sistemas embebidos, los FPGA se usan de dos maneras: ya sea para implementar las funcionalidades deseadas directamente con lógica digital, o utilizando la arquitectura de un microprocesador sea programada (*soft-core processor)* o empotrada (*hard-core processor)* y los periféricos deseados de un microcontrolador. Esta última opción se ha hecho muy popular en los últimos años, por lo que los precios del FPGA se han reducido significativamente y hoy en día pueden competir con microcontroladores. Los FPGA se usan con frecuencia para creación de prototipos y para reemplazó de tecnología en baja cantidad producción. Por lo anterior, la flexibilidad que el FPGA tiene respecto al desarrollo de sistemas lo hace una excelente alternativa para aplicaciones en las que los estándares están evolucionando, como la televisión digital, la electrónica de consumo, los sistemas de ciberseguridad, las comunicaciones inalámbricas, sistemas basados en la nube, entre otros.

3.3.5 Circuitos Integrados de Aplicación Específica (ASIC)

Un ASIC está diseñado específicamente para su aplicación prevista. Solo tiene los bloques necesarios para un funcionamiento óptimo, incluida una CPU, GPU, memoria, etc. Aunque los diseñadores pueden incorporar núcleos IP de terceros, como la CPU ARM Cortex, o bloques prediseñados para funciones estándar, como una capa física de Ethernet, un ASIC es un diseño desde cero. Para acelerar el proceso de diseño los ASIC se construyen a partir de la composición de las llamadas celdas estándar. Con los años, las herramientas de diseño han mejorado satisfaciendo la demanda de una mayor complejidad y funcionalidad de los sistemas. Los diseños actuales pueden incluir celdas estándar como: procesadores de hasta 32 bits, ROM, RAM, EEPROM, Flash y otros grandes bloques complejos. Es el más adecuado para aplicaciones de gran volumen. Por ejemplo, la unidad de procesamiento de tensor (TPU), es un acelerador desarrollado por Google específicamente para el aprendizaje automático de redes neuronales. Google también ofrece acceso TPU a empresas externas a través de su unidad de computación en internet [11].

3.4 Análisis de las Tecnologías para aplicaciones en tiempo real.

Los criterios que se utilizaron para evaluar las opciones y su importancia relativa son los siguientes:

- Tiempo de comercialización (TC) - Importancia media
- Rendimiento (R)- Alta importancia
- Precio (P) - Alta importancia
- Facilidad de desarrollo (FD) - Alta importancia
- Potencia (POT)- Importancia media
- Flexibilidad de funciones (FF) - Alta importancia
- Interfaz software-hardware (ISH) - Alta importancia

Para evaluar las tecnologías, se utilizó lo reportado en la literatura respecto a análisis realizados para aplicaciones en tiempo real [9, 18]. Las calificaciones cualitativas utilizadas en la evaluación son: Malo, Aceptable, Bueno y Excelente. Se consideraron los cambios que las diversas tecnologías han tenido en los últimos años, extrayendo la siguiente información:

Tabla 3.1: Análisis de Tecnologías Digitales.

	TC	R	P	FD	POT	FF	ISH	Resultado
MPU	Bueno	Bueno	Aceptable	Bueno	Aceptable	Excelente	Excelente	Bueno
MCU	Excelente	Aceptable	Excelente	Bueno	Aceptable	Excelente	Malo	Bueno
DSP	Excelente	Excelente	Bueno	Excelente	Excelente	Excelente	Aceptable	Excelente
FPGA	Bueno	Excelente	Excelente	Excelente	Bueno	Excelente	Bueno	Excelente
ASIC	Malo	Excelente	Excelente	Aceptable	Bueno	Malo	Aceptable	Aceptable

3.4.1 Discusión de las alternativas

3.4.1.1 MPU

Los MPU se consideran BUENOS en general. Ellos pueden utilizar la capacidad de programación del software para lograr diferentes funciones y características, ahorrando tiempo de comercialización frente a similares implementaciones lógica con hardware. Respecto al procesamiento de señal en tiempo real, MPU se califica como bueno, ya que cuenta con una amplia variedad de productos y herramientas. Sin embargo, no es fuerte con aplicaciones en tiempo real, ya que no esta tan orientado a aplicaciones empotradas, centrándose para fines de computadoras. Los MPU son buenos en materia de rendimiento y son la mejor alternativa para

interfaces software/hardware. Los niveles de la frecuencia de operación son típicamente en MHz y GHz, lo que produce un procesamiento de señal decente, pero la falta de instrucciones matemáticas específicas de un ciclo y las unidades multiplicadoras limitan su rendimiento en tiempo real. Con el uso adecuado de la programación de alto nivel y/o el uso de módulos de código estándar, se puede reducir significativamente el tiempo de desarrollo y, por lo tanto, ahorrar el costo de desarrollo. Sin embargo, aun cuando los diseñadores que trabajan con MPU's cuentan con soporte general, ellos no reciben ayuda por parte de los proveedores respecto a la aplicación específica o en tiempo real, debido a la falta de conocimiento de la misma lo que incrementa el tiempo de desarrollo [18].

3.4.1.2 MCU

Los MCU se consideran también BUENOS en general para procesamiento en tiempo real. Son los mejores en aspectos de comercialización y precio. Pueden utilizar la capacidad de programación del software para lograr diferentes funciones y características, ahorrando tiempo de comercialización frente a una lógica codificada similar implementaciones. Para el procesamiento de señal en tiempo real, MCU se clasifica como bueno ya que tiene una gama grande de productos y herramientas, aunque no tienen una buena combinación para aplicaciones de tiempo real. En comparación con los MPU, el MCU tiene menores recursos de procesamiento matemático y, por lo general, una frecuencia de operación más lenta. Por lo general, los MCU son de naturaleza general, lo que los hace menos eficiente en el consumo de energía que los DSP bajo situaciones similares. Sin embargo, generalmente usan menos recursos de silicio que los MPU y FPGA, lo que los hace más eficientes energéticamente que esas alternativas. Las alternativas de programación de los MCU's permite ciclos de desarrollo más rápidos para la función deseada, en lugar de tener que desarrollar chips específicos de la aplicación o ASICS. Con el uso adecuado de la programación de alto nivel y/o el uso de módulos de código estándar, el tiempo de desarrollo se puede reducir significativamente, ahorrando costo de desarrollo. En términos de ayuda para el desarrollo, la mayoría de los proveedores de MCU también tienen una red de soporte, aunque no tiene una puntuación excelente porque gran parte de esa red no tiene experiencia con aplicaciones en tiempo real, simplemente aplicaciones integradas [18].

3.4.1.3 DSP

DSP se consideró una EXCELENTE opción en general debido a el mejor y más relevante producto, conjunto de herramientas y soporte web para aplicaciones en tiempo real. El DSP tiene un excelente rendimiento, en particular, las arquitecturas multi-MAC que suelen tener ofrecen señales MIPS y MMACS muy altas rendimiento de procesamiento que resulta en una densidad creciente de canales por chip que es competitivo con un FPGA o un ASIC. El DSP no es tan rentable como ASIC o MCU por lo que su precio es más alto comparativamente. El DSP es considerado como excelente en materia de facilidad de desarrollo, ya que los proveedores de DSP tienen redes de terceros que agregan valor en todos los aspectos, desde la consultoría hasta software, plataformas de hardware para sistemas. Además, herramientas potentes y fáciles de usar, complementadas por redes de soporte técnico e ingenieros de aplicaciones que entienden el mundo en tiempo real están listos para ayudar a los clientes a lograr sus diseños. En términos de costo de desarrollo, la programación de los DSP's permite ciclos de desarrollo más rápidos para función deseada versus desarrollo de chips específicos de aplicaciones o ASICS. Con el uso adecuado de programación de alto nivel y/o uso de módulos de código estándar, se puede reducir el tiempo de desarrollo significativamente. Respecto a la flexibilidad de funciones puede utilizar la capacidad de programación del software para lograr diferentes funciones y características, ahorrando tiempo versus implementaciones lógicas codificadas similares [18].

3.4.1.4 FPGA

En general el FPGA es otra EXCELENTE alternativa para procesamiento. Aun cuando su tiempo de comercialización está por debajo de DSP, MCU y MPU (lo cual es de baja importancia en este caso), tienen mejor soporte y tiempo de ciclo más rápido que los ASIC, y por lo tanto puede reclamar tiempo de comercialización más rápido que esas alternativas. FPGA se consideró excelente en materia de rendimiento. Con el FPGA, los desarrolladores pueden configurar compuertas de hardware específicas para la aplicación, que ofrecen un alto rendimiento. Los avances tecnológicos han reducido la potencia de FPGA. La facilidad de desarrollo es excelente calificándolo como el mejor en costo de desarrollo asumiendo dos situaciones: que el conjunto de herramientas para la programación FPGA no es demasiado costoso; y, suponiendo que el desarrollador está tratando principalmente con hardware, que el ingeniero está involucrado en el desarrollo. Si el desarrollo se inclina hacia

ingenieros de software, entonces FPGA incrementa el esfuerzo y costo relativo pero ofrece una solución integral software/hardware. En términos de ayuda al desarrollo, las herramientas FPGA y la estructura de soporte para diseños basados en FPGA están bien establecidas y son aceptables. Dado que se puede reconfigurar para características adicionales o cambios. Sin embargo, este acto de reprogramación de hardware hoy en día compite con las soluciones programadas en software que ofrece un DSP [18].

3.4.1.5 ASIC

ASIC es clasificada como la opción menos recomendable para el desarrollo del sistema, ya el tiempo requerido en el ciclo de desarrollo es largo para completar y probar un ASIC, los diseños en obleas exclusivas y la validación de la solución puede tomar muchos meses o hasta años. Ya sea que se configure desde compuertas o elementos de procesador de nivel superior, solo son ligeramente diferentes en grado de dificultad. ASIC es excelente en materia de rendimiento. Tanto para un ASIC como para un FPGA, se pueden configurar compuertas hardware específicas para su aplicación que ofrecen un alto rendimiento. ASIC fue considerado excelente en materia de precios, ya que es la única alternativa para hacerlo. Además, la mercantilización ASIC puede reducir el precio por chips. El diseño está específicamente dirigido a la eficiencia energética, sin embargo, la mayoría de los diseños se centran en el rendimiento o el costo recurrente (precio), no en la potencia consumida. Aunque ASIC puede ser percibido como de bajo costo, de hecho es el más caro cuando se trata del costo total de desarrollo. Esto es atribuido a la cantidad de diseño lógico que debe realizar el cliente para crear la aplicación en el chip, junto con el costo cada vez mayor del procesamiento de silicio, con múltiples cientos de miles de dólares por revisión de retícula completa de silicio. En términos de ayuda al desarrollo, ASIC proporciona soporte general, pero no ofrece ninguna ayuda específica de la aplicación debido a falta de conocimiento del contenido específico de la misma por parte de los proveedores [18].

3.5 Elección: DSP vs FPGA

Derivado del análisis de tecnologías y en base a los resultados obtenidos, es necesario decidir el tipo de plataforma de hardware a utilizar. El tipo de sistema que se desarrollará, hoy en día generalmente se implementa en dos tipos de plataformas programables: los DSP o los FPGA. Como ya se describió anteriormente, los DSP son en cierta forma un microprocesador

de señal especializado, mientras que los FPGA son una forma de hardware altamente configurable.

Hace algunos años el uso de DSP era casi insustituible, pero las necesidades de muchas aplicaciones y su interacción con el usuario (Interfaces Gráficas de Usuario, GUI) han superado las capacidades de procesamiento (medidos en millones de instrucciones por segundo, MIPS)y disponibilidad de herramientas software, esto ha dado una oportunidad al uso de los FPGA's. Actualmente, la razón principal por la que se opta por usar un FPGA sobre un DSP es por los requisitos MIPS de la aplicación. Por lo tanto, la comparación entre el DSP y el FPGA se centran en la comparación de MIPS, que, si bien es importante, no es la única ventaja que el FPGA tiene, ya que de igual importancia, y a menudo pasada por alto, la fiabilidad y mantenibilidad son otra ventaja que tienen los productos basado en FPGA [19].

En el ámbito de aplicación la mayoría de los proyectos de ingeniería están sujetos a actualizaciones o lanzamiento de software de los productos. En la mayoría de las empresas de tecnología, generalmente hay una larga lista interna de errores de software o informes de problemas junto con versiones de software que contendrán los parches o arreglos asociados. Es común esperar que todo el software, incluido el código DSP, contenga cierto nivel de errores y lo mejor que se puede hacer es minimizarlos. Ya que el DSP en esencia se reconfigura constantemente para muchas tareas diferentes, algunas de ellas relacionados con señales y otros más orientados al control, los recursos como los registros centrales del procesador, internos y la memoria externa, los controladores DMA y los periféricos de E/S son compartidos por todas las tareas, a menudo denominadas "subprocesos". Estas crean muchas oportunidades para que las tareas interactúen, a menudo de maneras inesperadas o no obvias.

Por otro lado, los diseños con FPGA tienden a actualizarse con mucha menos frecuencia y generalmente es un evento inusual para un fabricante emitir una actualización de un archivo de configuración FPGA. Las razones detrás de esto se deben a las diferencias en el proceso de desarrollo de ingeniería entre el DSP y el FPGA.

Por todo lo anterior, a continuación se presentan las ventajas que, para nuestro sistema tiene el utilizar un DSP o un FPGA [20]:

DSP

- Arquitectura optimizada para computar algoritmos DSP.
- Excelente compensación MIP/mW/costo.
- Compiladores eficientes: puede programar toda la aplicación en C.
- Algunos tienen un O/S en tiempo real (para la programación de tareas).
- Puede ser de muy baja potencia.
- Excelente para procesamiento de señal.

FPGA

- Son los más rápidos para efectuar cálculos computacionales.
- Excelentes herramientas de soporte para el diseño.
- Variedad de dispositivos generalmente requerido en el diseño.
- Capacidad para sintetizar cualquier periférico.
- Fácil de desarrollar con funciones flexibles: reprogramables en campo.
- Excelente para el procesamiento de señales paralelas y de alta velocidad.
- Disponibilidad con procesadores integrados para uso con SO gráfico.

Por todo lo anterior y dado que el sistema a desarrollar estará más orientado a generación de señales, más que a procesamiento, y adicionalmente que requiere una interfaz gráfica para el control del hardware, el sistema se desarrollará en un FPGA.

Capítulo 4

Generación de Barridos para Ondas Interferenciales.

En este capítulo se presentan los algoritmos utilizados en la generación de barridos con ondas interferenciales. Se utiliza la herramienta computacional MatLab para las pruebas en simulación de la obtención de los coeficientes para generación de señales. Con los datos obtenidos en simulación, se realiza las arquitecturas hardware en un FPGA correspondientes al generador de ondas Kotz.

Como se mencionó en el capítulo 2, la generación de barridos se realiza utilizando modulación en amplitud de una onda base que es una función senoidal de 2,500 Hz y mediante el control de barridos en tiempo, ecuación 2.18, es posible generar señales interferenciales acorde a los parámetros establecidos en la electroterapia. A continuación se describen algunos efectos que los barridos a frecuencias específicas tiene en el musculo, posteriormente se presentan resultados del algoritmo desarrollado en MatLab para la generación de las señales y finalmente la realización del sistema generador de señales basado en un FPGA

4.1 Efectos de las frecuencias en el músculo

Diversos equipos comerciales [21, 22] incluyen programas preestablecidos, así como los efectos que las frecuencias utilizadas tienen sobre el músculo:

1Hz −3Hz: Efecto descontracturante y relajante, se aplica en molestias o dolores ocasionados por contracturas 4Hz − 7Hz: Incremento de la segregación de endorfinas y encefalinas, provocando una anestesia local en 5Hz acompañada de la disminución de la ansiedad. Así como también facilitando el sueño a 7Hz y el aumento del flujo sanguíneo con hiperoxigenación. Generalmente se aplica para evitar los calambres, reoxigenar los tejidos, acelerar el retorno venoso, la eliminación de edemas y metabolitos acumulados.

8Hz−10Hz: Efecto del aumento del flujo sanguíneo por cinco veces aproximadamente 10Hz−33Hz: Efecto para el aumento muscular y su resistencia localizada, se aplica para abdominales y glúteos, a su vez la grasa corporal es utilizada como mecanismo de energía.

33Hz−50Hz: Efecto del aumento del tono muscular sin desarrollo de la musculatura, se aplica generalmente en glúteos, adductores y abdominales con fines estéticos.

50Hz−75Hz: Efecto del aumento de la fuerza y resistencia localizada. Al ser combinadas con el entrenamiento voluntario en las mismas sesiones, proporciona un aumento muscular y preserva las articulaciones.

75Hz−150Hz: Efecto de supratetanización de las fibras, es decir que está sobre la contracción repetida de un músculo a una frecuencia, por encima de la frecuencia de la fusión tetánica

debida a la estimulación del mismo o del nervio correspondiente, no permitiendo la relajación entre las distintas contracciones. La intensidad de corriente eléctrica comienza con 28mA−120mA o la máxima que se pueda soportar.

4.2 Parámetros de los Barridos

La función de prueba trapezoidal a generar se muestra en la Figura 2.6. Los parámetros necesarios para la generación son:

T_{FS} : Tiempo del flanco de subida.
T_C: Tiempo de contracción muscular.
T_{FB}: Tiempo del flanco de bajada.
T_R: Tiempo de relajación muscular.

Como se mencionó en el Capítulo 2, las corrientes interferenciales son alternas o bifásicas y simétricas en la gama de frecuencia media, es decir, que están comprendidas desde 2, 000Hz − 10, 000Hz. Son de tipo senoidal, particularmente con una frecuencia de 2, 500 Hz (frecuencia media). Para aplicarlas en los músculos estas ondas no deben ser continuas Figura 4.1, sino que se deben producirse en forma de ráfagas, es decir, contracción y relajación presentes y ausentes a un ritmo predeterminado.

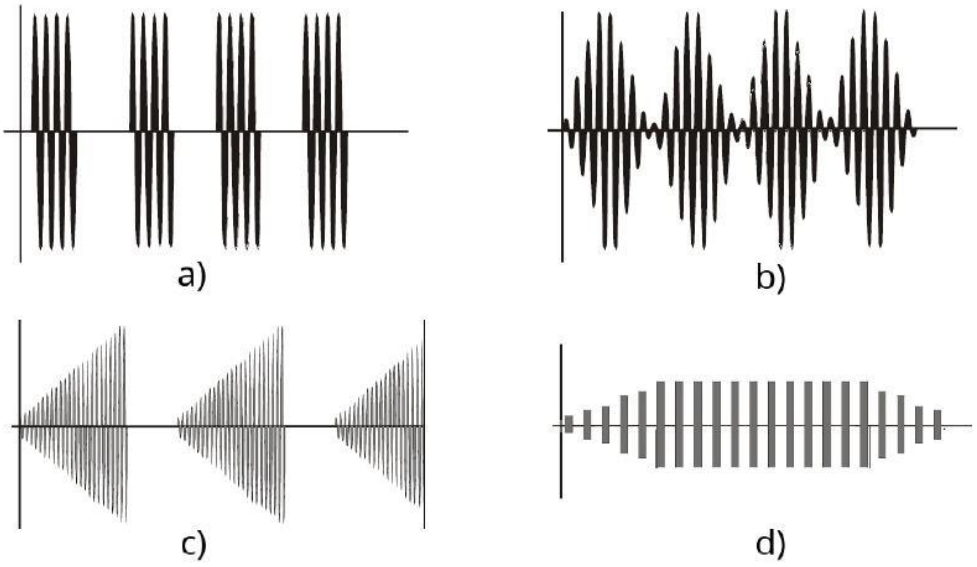

Figura 4.1: Contornos para la modulación y barridos de las ondas Rusas: a) cuadrangular, b) senoidal, c) triangular y d) trapezoidal.

4.3 Generación de barridos utilizando Matlab

Para la generación de barridos, se programa el algoritmo de la Figura 4.2 utilizando la herramienta de MatLab [23], la ecuación 2.16, para un barrido trapezoidal y la ecuación 2.17 para un barrido cuadrado. Las Figuras 4.3 y 4.4 muestran los resultados obtenidos:

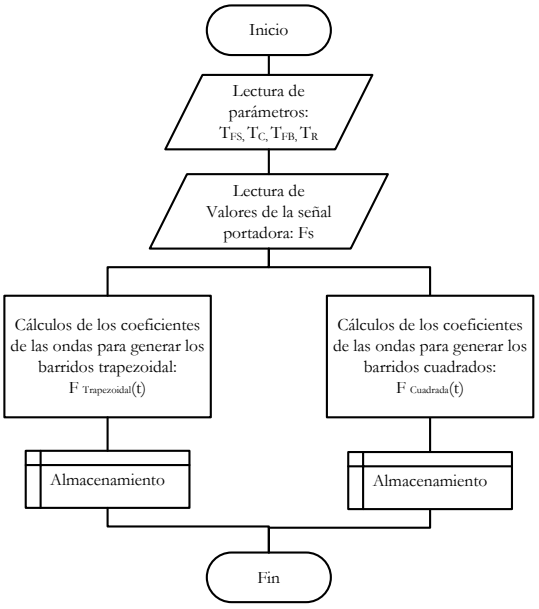

Figura 4.2: Diagrama de flujo del algoritmo para generación de barridos de ondas rusas. Los parámetros de T_{FS}, T_C, T_{FB}, T_R corresponden a los tiempos para la función trapezoidal. Fs es la frecuencia de muestreo para la señal de salida.

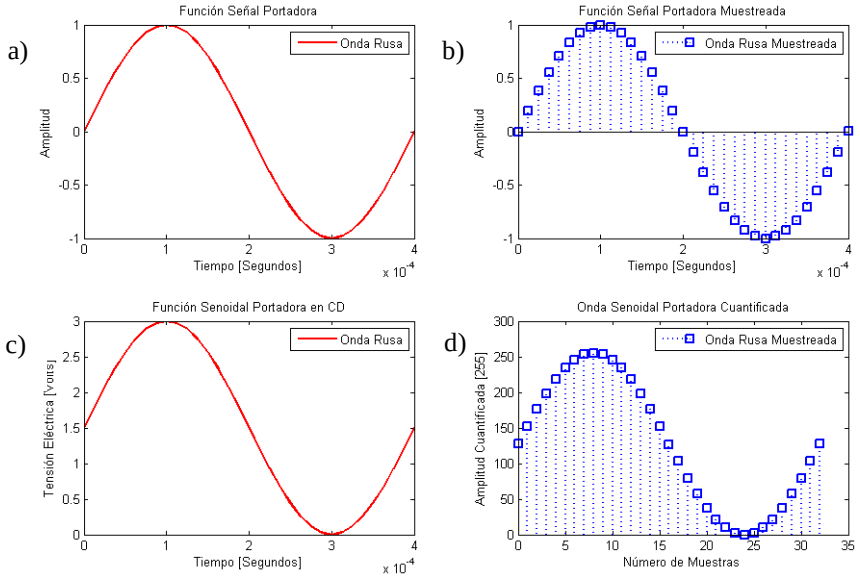

Figura 4.3: Acondicionamiento de la señal portadora para la generación de los barridos de las ondas rusas. a) Señal base generada con funciones de Matlab. b) Discretización de la señal base. c) Adición de un off-set a la señal base. d) Señal discretizada y cuantificada de la señal portadora.

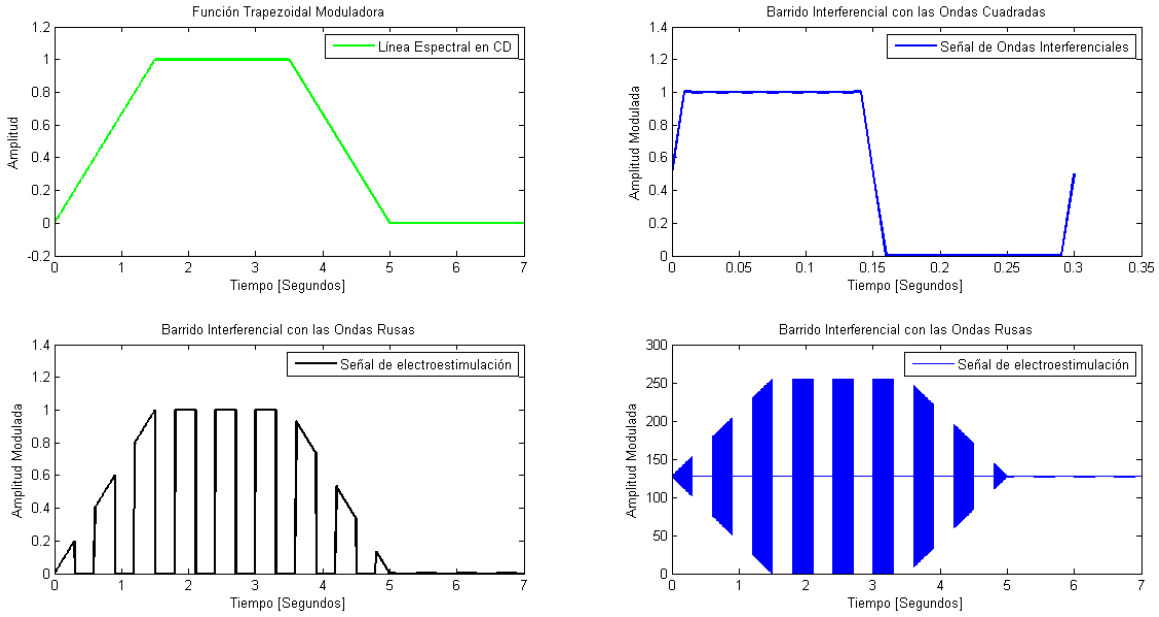

Figura 4.4: Barridos generados en base a los parámetros de las señales moduladoras. T_{FS}=1,5s, T_C=2s, T_{FB}=1,5s, T_R=2s. Fs= 32 muestras.

Una vez validado el comportamiento de los barridos, con los valores obtenidos en la simulación, se procede a utilizarlos para la implementación en el FPGA.

4.4 Arquitectura para la generación de barridos en Hardware.

Para el desarrollo del sistema en hardware, se utilizó una tarjeta FPGA Max 10. Se utilizó el software Quartus II (Figura 4.5) de la empresa Intel-FPGA en su versión académica [24]. En el desarrollo de la arquitectura, se utilizaron los valores obtenidos en la simulación en MatLab. Dado que los valores son con punto flotante (enteros y fracciones), se realizó un proceso de escalamiento, con la finalidad de transformarlos a enteros (punto fijo) y de esta forma sea posible utilizarlos en el FPGA (ya que este dispositivo no reconoce fracciones).

En general el diseño está conformado de 4 bloques, como se muestra en la Figura 4.6. Los elementos principales son un contador (Contador_Binario), para direccionamiento de la memoria ROM donde se encuentran los valores codificados en punto fijo; una memoria ROM con los datos de la señal para generar barridos; un divisor de frecuencia para atenuar los ciclos por segundo de la señal de salida y un contador de 23 horas para sincronizar el sistema con

programación de rutinas de hasta un día. El lenguaje de programación utilizado es VHDL (acrónimo de la combinación de: VHSIC *Very High Speed Integrated Circuit* y HDL *Hardware Description Language*) y se utilizan las librerías estándar (por ejemplo: IEEE.numeric_std) en la implementación.

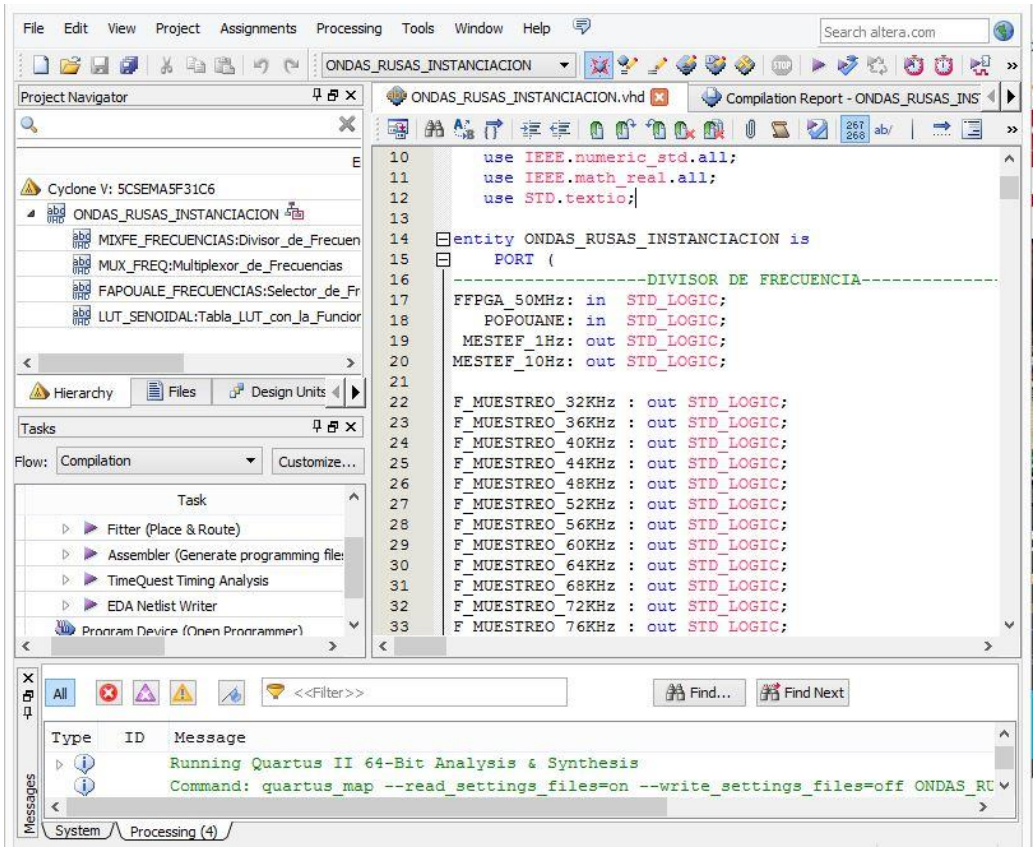

Figura 4.5: Entorno de programación de dispositivos FPGA, propiedad de la empresa Intel-FPGA. La versión académica permite realizar todo tipo de desarrollos, la diferencia con la versión comercial es que no cuenta con cores (núcleos de hardware) disponibles, por lo que, las arquitecturas se tienen que desarrollar casi desde cero.

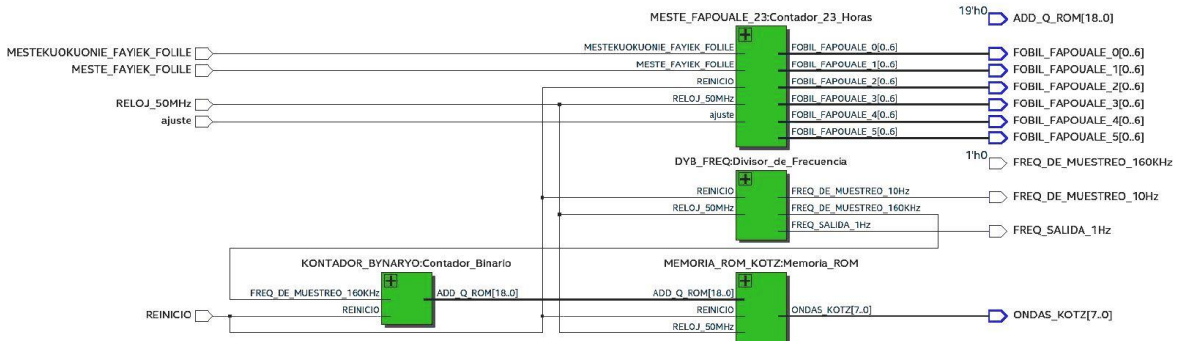

Figura 4.6: Arquitectura del sistema generador de ondas interferenciales.

La señal generada por el sistema puede observarse en la Figura 4.6, donde a través del uso de un osciloscopio, puede observarse la señal que el sistema genera en tiempo real.

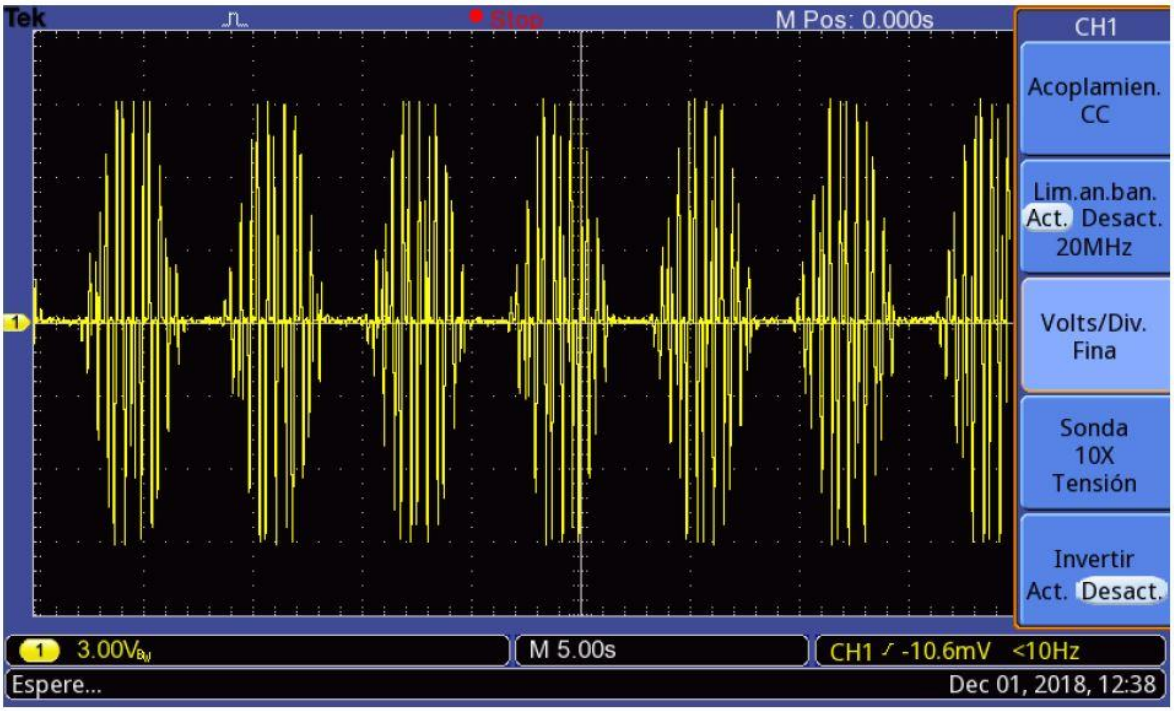

Figura 4.6: Corrientes interferenciales con Ondas Rusas. Se generan barridos de 5Hz con una frecuencia de la portadora de 2,5 kHz.

4.5 Generación de interfaz software para el control de los barridos.

Una vez que se validó la operación de la arquitectura hardware, el entorno de programación de Quartus II, tiene herramientas como el Qsys (Figura 4.7) que permite incorporar las arquitecturas desarrolladas a sistemas que incluyan un procesador, con la finalidad de poder utilizar programación de alto nivel (como lenguaje C) para el control y manipulación del hardware.

Una vez desarrollado el sistema con esta herramienta y al incorporarle un procesador, es este caso un soft-core (procesador programado con lenguaje HDL, llamado NIOS2) la interacción con el hardware se realiza a través de un compilador de lenguaje C. El entorno de Quartus II tiene incorporado una interfaz de programación llamada Eclipse, Figura 4.8, para el control de las arquitecturas diseñadas, pero no se está restringido a que el acceso al NIOS2

solo sea a través de esta, sino que cualquier compilador de C puede accesar al hardware utilizando las librerías del sistema NIOS2.

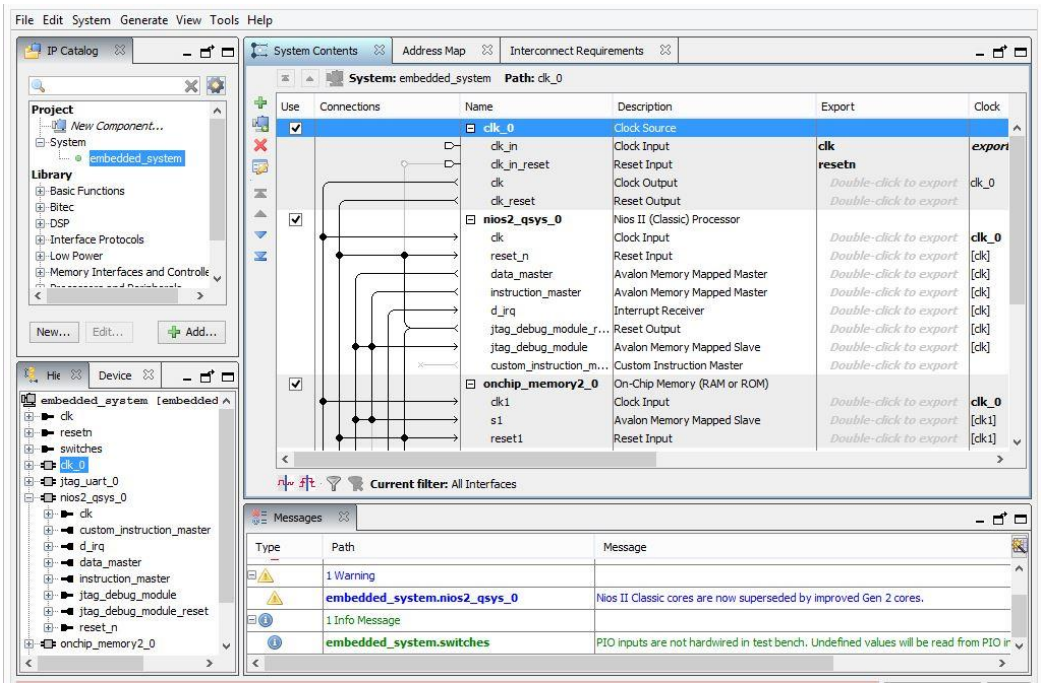

Figura 4.7: Entorno de programación con Qsys. Esta herramienta por default considera un sistema secuencial basado en un microprocesador, permitiendo el uso de software en la manipulación de señales.

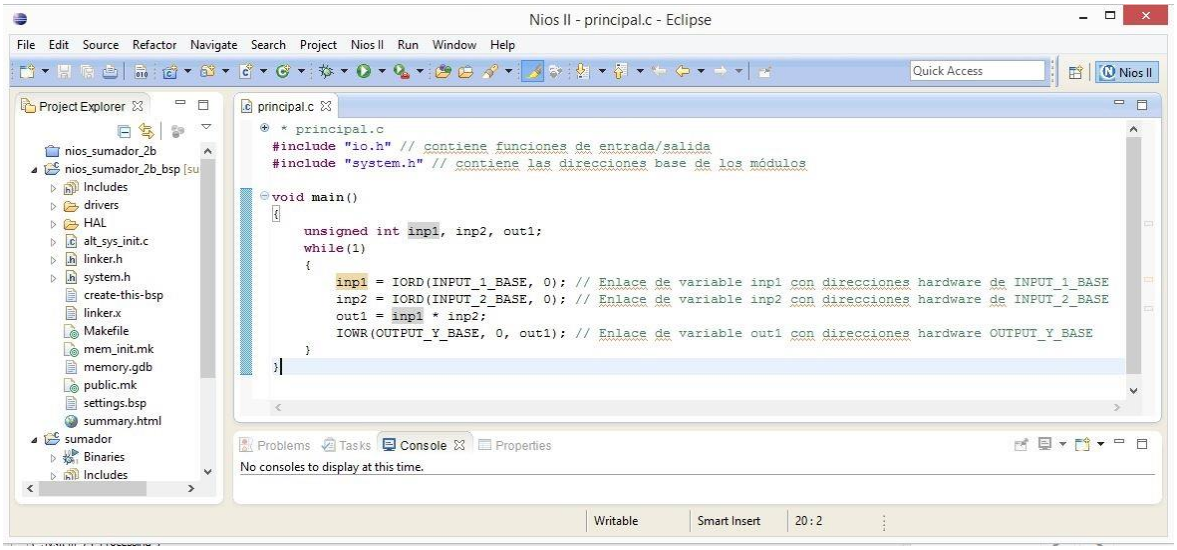

Figura 4.8: Entorno de programación con lenguaje de alto nivel incorporado en el Quartus II. La herramienta se llama Eclipse e incorpora un compilador de lenguaje C. El enlace entre las variables software y los elementos hardware de la arquitectura se realiza a través de librerías que contienen las direcciones base de los elementos del sistema en el FPGA.

Utilizando software para desarrollo de entornos gráficos como Java, fue posible realizar la programación de una interfaz de control del hardware. En las pruebas realizadas se utilizaron diversos componentes para la generación de las señales y sus parámetros. La Figura 4.9, muestra una interfaz básica desarrollada para el control de algunos parámetros de la señal a generar por el sistema.

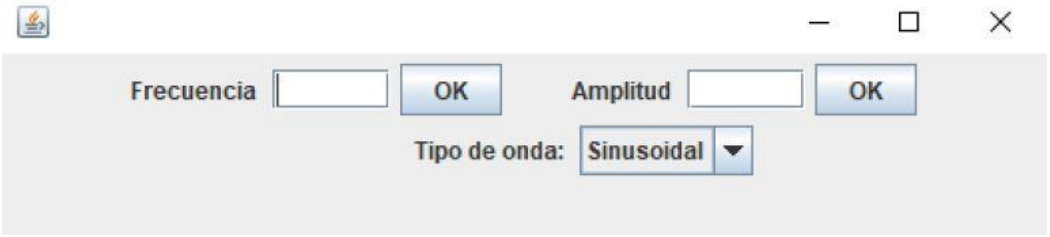

Figura 4.9: Interfaz gráfica realizada en Java para el control de la arquitectura hardware del sistema.

Conclusiones

En esta sección se realiza una discusión de los resultados obtenidos con las arquitecturas hardware diseñadas. Además, se mencionan las principales contribuciones que este trabajo tuvo en la generación de señales con hardware reconfigurable. Finalmente, se hace una serie de propuestas a futuro en esta línea de desarrollo.

5.1 Discusión.

El propósito de este trabajo fue el desarrollo de un sistema capaz de generar barridos de señales interferenciales, con la finalidad de que posteriormente el prototipo pueda utilizarse como equipo para terapias física utilizando electroestimulación. La realización del sistema fue motivada por varios aspectos:

1. ***Arquitecturas de propósito específico:*** cuya realización permitirá generar diversos barridos a diferentes frecuencias base, flexibilidad que no ofrecen lo equipos comerciales.
2. ***Aplicaciones de los dispositivos reconfigurables en el área médica:*** lo que permite utilizar esta tecnología para sistemas que requieren gran fiabilidad en cuanto a las señales que se generan, derivado de que cualquier anomalía puede perjudicar o poner incluso en riesgo los órganos del cuerpo humano.
3. ***El factor económico:*** ya que los equipos comerciales en el mercado tienen un alto costo, lo que limita la disponibilidad de ellos en hospitales y clínicas de acceso general.

En el proceso de desarrollo se llevaron a cabo las siguientes tareas:

1. Desarrollo de la matemática necesaria para la generación de señales y barridos.
2. Simulación de los modelos matemáticos y generación de valores para la implementación de los barridos en hardware.
3. Realización de pruebas en diferentes tecnologías digitales, para elegir las más conveniente para el sistema.
4. Desarrollo de una interfaz software gráfica con la que se pueda controlar el sistema de generación de señales.

Cada una de estas tareas requirió de un análisis, realización de pruebas, validación de resultados y ajustes hasta llegar a los valores aceptables (tomando como referencia los parámetros y características de equipos comerciales). Para ello se propuso un sistema reconfigurable basado en módulos hardware interconectados entre sí. Cada arquitectura de procesamiento (módulo) requirió en su realización de estrategias específicas debido a la exactitud requerida versus demanda de recursos disponibles de cada etapa.

Para la evaluación de la exactitud de los resultados, se utilizaron tanto funciones predefinidas en software (Toolbox de MatLab) como equipos medidores de señal (osciloscopio). Dado que no se cuenta con un equipo comercial, no fue posible comparar la operación real del sistema con uno de ellos, sin embargo, acorde a los parámetros registrados por el osciloscopio, si corresponden en tiempo y valores de frecuencia a los establecidos con los parámetros de entrada.

Finalmente, derivado de contratiempos técnicos, no fue posible el integrar como un solo sistema el software y hardware, sin embargo se generaron los bloques funcionales para cada parte, lo que se deja como trabajo a futuro para la conclusión del proyecto

5.2 Contribuciones del trabajo.

Las contribuciones de este proyecto pueden ser resumidas en los siguientes puntos:

- En el área de modelado de señales, se obtuvieron y validaron de forma práctica las ecuaciones para generar barridos de señales.
- Se han desarrollado arquitecturas para la generación de señales base y modulaciones en amplitud.
- Se validó el proceso de integrar software para controlar hardware mediante pruebas de comunicación.
- Se desarrolló una interfaz gráfica sencilla para la manipulación de las señales que el sistema genera.

5.3 Conclusión.

Los resultados presentados en este trabajo muestran que el desarrollo de los diferentes módulos del sistema: arquitectura para generación de señales e interfaz software para control, son funcionales, es decir, trabajan acorde a los parámetros establecidos según la literatura revisada y fueron verificados con equipos de metrología eléctrica.

Es importante aclarar que queda pendiente la integración en una sola tarjeta de todo el sistema, derivado de requerimientos técnicos específicos para lograrlo como lo son el que el FPGA tenga empotrado un procesador donde se pueda trabajar con el software desarrollado.

5.4 Trabajo a Futuro.

El proyecto realizado conllevo la revisión de modelos matemáticos, programación hardware y programación software. En las tres áreas se tuvo un avance significativo (prácticamente al 90%), quedando pendiente solo el ajuste de algunos parámetros, circuitos y botones software para que se pueda llevar a cabo la integración en un solo sistema de todo lo desarrollado. Adicionalmente, queda pendiente la realización de una validación general acorde a los estándares internacionales para este tipo de sistemas.

Finalmente como recomendación, se sugiere utilizar una tarjeta con microprocesador empotrado, tecnología SoC (System on a Chip) que facilita la interacción software-hardware.

Apéndice A

Códigos de simulaciones y arquitecturas.

En este apéndice se presentan los códigos realizado en el proyecto.

C.1 Generación de Barridos en MatLab.

```
clc, clear all; close all;
disp('=============================================O====================================');
disp('    PROGRAMA EN MATLAB R2018a PARA GENERACIÓN DE BARRIDOS CON ONDAS KOTZ (RUSAS)    ');
disp('                           Por: Clemente C. C., ');
disp('==================================================================================');

% MÓDULO DE INTRODUCCIÓN DE LOS PARÁMETROS DE LA FIGURA DEL TRAPEZOIDE

disp('PARÁMETROS REQUERIDOS PARA LA FUNCIÓN TRAPEZOIDAL COMO BARRIDOS.');
tfs  = input('INTRODUZCA EL TIEMPO DE FLANCO DE SUBIDA [SEGUNDOS]: ');
tC   = input('INTRODUZCA EL TIEMPO DE CONTRACCIÓN MUSCULAR [SEGUNDOS]: ');
tfb  = input('INTRODUZCA EL TIEMPO DE FLANCO DE BAJADA [SEGUNDOS]: ');
tR   = input('INTRODUZCA EL TIEMPO DE RELAJACIÓN MUSCULAR [SEGUNDOS]: ');
disp('----------------------------------------------------------------------------------');
disp('PARÁMETROS REQUERIDOS PARA LA FUNCIÓN CUADRADA COMO INTERFERENCIALES.');
PCuad = input('INTRODUZCA EL PERIODO DE LA ONDA CUADRADA, RÁFAGAS [MILISEGUNDOS]: ');
disp('----------------------------------------------------------------------------------');
MRusa = input('INTRODUZCA LA CANTIDAD DE MUESTREOS DE LA ONDA RUSA
Muestreo:16.32.64,128,256]: ');
ITER  = input('INTRODUZCA LA CANTIDAD DE ITERACIONES [n=100-1000]: ');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% MÓDULO DE CÁLCULOS PRIMORDIALES DE LOS PARÁMETROS PARA EL TRAPEZOIDE

D1 = tfs; %%%%%%%%%%%%%%%%%% TIEMPO DE D1
D2 = (D1) + (tC); %%%%%%%%%% TIEMPO DE D2
D3 = (D2) + (tfb); %%%%%%%%% TIEMPO DE D3
D4 = (D3) + (tR); %%%%%%%%%% TIEMPO DE D4
tfs1 = D1; %%%%%%%%%%%%%%%%% TIEMPO PARA EL FLANCO DE SUBIDA
tC1 = (D2) - (D1); %%%%%%%%% TIEMPO PARA LA CONTRACCIÓN MUSCULAR
tfb1 = (D3) - (D2); %%%%%%%% TIEMPO PARA EL FLANCO DE BAJADA
tR1 = (D4) - (D3); %%%%%%%%% TIEMPO PARA LA RELAJACIÓN MUSCULAR
PTrapezio = D4; %%%%%%%%%%%% PERIODO TOTAL DE LA FUNCIÓN TRAPEZOIDAL
FTrapezio = 1/PTrapezio; %%% FRECUENCIA TOTAL DE LA FUNCIÓN TRAPEZOIDAL
WTrapezio = 2*(pi)*(FTrapezio); %%% FRECUENCIA ANGULAR DE LA FUNCIÓN TRAPEZOIDAL
A = 1; %%%%%%%%%%%%%%%%%%%%% AMPLITUD UNITARIA DE LA TENSIÓN ELÉCTRICA

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% MÓDULO DE CÁLCULOS PRIMORDIALES DE LOS PARÁMETROS PARA LA FUNCIÓN CUADRADA

PCuadrada = PCuad/1000; %% PERIODO DE LA FUNCIÓN CUADRADA EN MILISEGUNDOS
FCuad = 1/PCuadrada; %%%%% FRECUENCIA DE LA FUNCIÓN CUADRADA
WCuad = 2*(pi)*(FCuad); %% FRECUENCIA ANGULAR DE LA FUNCIÓN CUADRADA
tCuadrada = 0:(PCuadrada/MRusa):PCuadrada; % INTERVALO DE TIEMPOS 0:(MUESTRAS):PERIODO ONDA
CUADRADA
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% MÓDULO DE CÁLCULOS PRIMORDIALES DE LOS PARÁMETROS PARA LA ONDA RUSA

FSenoidal = 2500 ; %%%%%%%%%%%%%%%%%% FRECUENCIA DE OSCILACIÓN PARA LAS ONDAS KOTZ, 2,500Hz
PKOTZ = 0.0004 ; %%%%%%%%%%%%%%%%%%%% PERIODO DE LAS ONDAS DE KOTZ, 0.0004 SEGUNDOS
WSenoidal = 2*(pi)*(FSenoidal) ; %%%% FRECUENCIA ANGULAR DE LAS ONDAS KOTZ
MUESTMIN = (PKOTZ)/MRusa ; %%%%%%%%%% MUESTRA MÍNIMA DE LA FUNCIÓN TRAPEZOIDAL
tMuestreo = 0:(MUESTMIN):PTrapezio; % TIEMPO-MUESTREO: FUNCIÓN TRAPEZOIDAL, CUADRADA Y
SENOIDAL
FMuestreoSenoidal = 2*2500*MRusa;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% MÓDULO DE CÁLCULOS PRIMORDIALES PARA LA FUNCIÓN PORTADORA SENOIDAL

tKOTZ = 0:MUESTMIN:0.0004; %%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MUESTRAS DEL PERIODO DE 0-0.0004
MuestreoORusa = 0:1:MRusa; %%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CANTIDAD DE MUESTREOS DE LA ONDA
RUSA
ORusa = (sin(WSenoidal*tKOTZ)); %%%%%%%%%%%%%%%%%%%%%%%%% ONDA RUSA PRINCIPAL
ORusaCD = 1.5+1.5*(sin(WSenoidal*tKOTZ)); %%%%%%%%%%%%%%% ONDA RUSA CON OFFSET
ORusaCDM = round(255*(0.5+0.5*(sin(WSenoidal*tKOTZ)))); % ONDA RUSA CON OFFSET REDONDEADA Y
MUESTREADA

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% MÓDULO DE CÁLCULOS PRIMORDIALES PARA LA FUNCIÓN TRAPEZOIDAL

% CÁLCULO DE LA CONSTANTE Bn
Bn = 0;
for n= 1:ITER
    Bn = Bn + ((2*A)/(PTrapezio*((n*(WTrapezio))^2))) * (((D3-D2)*(sin(n*(WTrapezio)*D1)) +
D1*(sin(n*(WTrapezio)*D2) - sin(n*(WTrapezio)*D3))) / (D1*(D3-D2))) *
(sin(n*(WTrapezio)*tMuestreo));
End

% CÁLCULO DE LA CONSTANTE An
An = 0;
for m= 1:ITER
    An = An + ((2*A)/(PTrapezio*((m*(WTrapezio))^2))) * (((D3-D2)*(cos(m*(WTrapezio)*D1) -
1) + D1*(cos(m*(WTrapezio)*D2) - cos(m*(WTrapezio)*D3))) / (D1*(D3-D2))) *
(cos(m*(WTrapezio)*tMuestreo));
end

COEFBn = Bn;
COEFAn = An;
TRAPEZOIDAL = ((A*(D3 + D2 - D1))/(2*PTrapezio)) + COEFAn + COEFBn;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% MÓDULO DE CÁLCULOS PRIMORDIALES PARA LA FUNCIÓN CUADRADA

CUAD = 0;
for l= 1:ITER
    CUAD = CUAD + ((sin((2*l - 1)*WCuad*tCuadrada))/(2*l - 1));
end
CUADRADA = abs((1/2) + (2/(pi))*(CUAD));

%-------------------------------------------------------------------------------------------

tCMuestreo = tMuestreo/2; %% FRECUENCIA DIVIDIDA ENTRE 2 DE LA ONDA CUADRADA
CUADINTER = 0;
for k= 1:ITER
    CUADINTER = CUADINTER + ((sin((2*k - 1)*WCuad*(tCMuestreo)))/(2*k - 1));
end
FCUADRADA = abs(round((1/2) + (2/(pi))*(CUADINTER)));
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% MÓDULO DE CÁLCULOS PRIMORDIALES PARA LAS SEÑALES INTERFERENCIALES EN AMPLITUD MODULADA

FINTERFERENCIAL = abs((TRAPEZOIDAL.*FCUADRADA)); %% FUNCIÓN INTERFERENCIAL EN AMPLITUD MOD.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% MÓDULO DE CÁLCULOS PRIMORDIALES PARA LA AMPLITUD MODULADA

ORusaPortadora = (sin(WSenoidal*tMuestreo)); % ONDA RUSA CON OFFSET REDONDEADA Y MUESTREADA
OINTERFERDEC = round(abs(255*(((1+((FINTERFERENCIAL).*(ORusaPortadora))))/2)));
OINTERFERBIN = dec2bin(OINTERFERDEC); %%%%%%%% CONVERSIÓN DECIMAL A BINARIO
OINTERFERHEX = dec2hex(OINTERFERDEC); %%%%%%%% CONVERSIÓN DECIMAL A HEXADECIMAL
OINTERDEC = [tMuestreo;OINTERFERDEC];
OINTERHEX = [OINTERFERHEX];
NVALORES = numel(OINTERFERDEC);

% MÓDULO DE GRÁFICAS DE LA FUNCIÓN PORTADORA SENOIDAL EN CA

figure(1);
subplot(2,2,1);
plot(tKOTZ,ORusa,'red','LineWidth',2);
legend('Onda Rusa');
title('Función Señal Portadora');
xlabel('Tiempo [Segundos]');
ylabel('Amplitud');

subplot(2,2,2);
stem(tKOTZ,ORusa,':bs','LineWidth',1.5);
legend('Onda Rusa Muestreada');
title('Función Señal Portadora Muestreada');
xlabel('Tiempo [Segundos]');
ylabel('Amplitud');

subplot(2,2,3); % ONDA RUSA EN CD
plot(tKOTZ,ORusaCD,'red','LineWidth',2);
legend('Onda Rusa');
title('Función Senoidal Portadora en CD');
xlabel('Tiempo [Segundos]');
ylabel('Tensión Eléctrica [Volts]');

subplot(2,2,4); % ONDA RUSA EN CD CUANTIFICADA
stem(MuestreoORusa,ORusaCDM,':bs','LineWidth',1.5);
legend('Onda Rusa Muestreada');
title('Onda Senoidal Portadora Cuantificada');
xlabel('Número de Muestras');
ylabel('Amplitud Cuantificada [255]');

figure (2);
subplot(2,2,1); % GRÁFICA DE LA FUNCIÓN TRAPEZOIDAL
plot(tMuestreo,TRAPEZOIDAL,'green','LineWidth',2);
legend('Línea Espectral en CD');
title('Función Trapezoidal Moduladora');
xlabel('Tiempo [Segundos]');
ylabel('Amplitud');

subplot(2,2,2); % ONDAS CUADRADAS
plot(tCuadrada,CUADRADA,'blue','LineWidth',2);
legend('Señal de Ondas Interferenciales');
title('Barrido Interferencial con las Ondas Cuadradas');
xlabel('Tiempo [Segundos]');
ylabel('Amplitud Modulada');

subplot(2,2,3); % ONDAS RUSAS
plot(tMuestreo,FINTERFERENCIAL,'black','LineWidth',2);
legend('Señal de electroestimulación');
title('Barrido Interferencial con las Ondas Rusas');
xlabel('Tiempo [Segundos]');
ylabel('Amplitud Modulada');
```

```
subplot(2,2,4); % ONDAS RUSAS
plot(tMuestreo,OINTERFERDEC,'blue','LineWidth',0.5);
legend('Señal de electroestimulación');
title('Barrido Interferencial con las Ondas Rusas');
xlabel('Tiempo [Segundos]');
ylabel('Amplitud Modulada');
```

C.2 Conversión de Formatos MatLab.

```
disp('=============================================O=====================================');
disp('           PROGRAMA PARA CONVERSIÖN DE VALORES PUNTO FLOTANTE A PUNTO FIJO'         );
disp('                              Por: Clemente C. C., ');
disp('=============================================================================');

n=8; D=2^n; e=8;
for i=1:D
    S(i)=sin(2*(i-1)*pi/D);
end;
for i=1:D
    Auxiliar = i-1;
    for j=n:-1:1
        X(i,j)= rem(Auxiliar,2);
        Auxiliar = floor(Auxiliar/2);
    end;
end;
for i=1:D
    Auxiliar = floor((1.0+S(i))*2^(e-1));
    if (Auxiliar==2^e)
        Auxiliar = Auxiliar-1;
    end;
    for j=e:-1:1
        Y(i,j)=rem(Auxiliar,2);
        Auxiliar=floor(Auxiliar/2);
    end;
end;
fid=fopen('LUT_seno.vhd','wt');
fprintf(fid,'library IEEE;\n');
fprintf(fid,'use IEEE.std_logic_1164.all;\n');
fprintf(fid,'\n');
fprintf(fid,'entity LUT_seno is\n');
fprintf(fid,'   port(\n');
fprintf(fid,'      F : in  std_logic_vector(%d downto 0);\n',n-1);
fprintf(fid,'      S : out std_logic_vector(%d downto 0)\n',e-1);
fprintf(fid,'      );\n');
fprintf(fid,'   end LUT_seno;\n');
fprintf(fid,'\n');
fprintf(fid,'architecture Tabla of LUT_seno is\n');
fprintf(fid,'begin\n');
fprintf(fid,'   process(F)\n');
fprintf(fid,'   begin\n');
fprintf(fid,'      case F is\n');
for i = 1:D
    fprintf(fid,'         when "');
    for j=1:n
        fprintf(fid,'%d',X(i,j));
    end;
    fprintf(fid,'" => S <= "');
    for j=1:e
        fprintf(fid,'%d',Y(i,j));
    end;
    fprintf(fid,'"; -- Argumento %d Funcion %1.8f\n',i-1,S(i));
end;
fprintf(fid,'         when others => null;\n');
fprintf(fid,'      end case;\n');
fprintf(fid,'   end process;\n');
fprintf(fid,'end Tabla;\n');
fclose(fid);
```

C.3 Valores para ondas.

```
library IEEE;
use IEEE.std_logic_1164.all;

entity LUT_seno is
   port(
      F : in  std_logic_vector(7 downto 0);
      S : out std_logic_vector(7 downto 0)
      );
   end LUT_seno;

architecture Tabla of LUT_seno is
begin
   process(F)
   begin
      case F is
         when "00000000" => S <= "10000000"; -- Argumento 0 Funcion 0.00000000
         when "00000001" => S <= "10000011"; -- Argumento 1 Funcion 0.02454123
         when "00000010" => S <= "10000110"; -- Argumento 2 Funcion 0.04906767
         when "00000011" => S <= "10001001"; -- Argumento 3 Funcion 0.07356456
         when "00000100" => S <= "10001100"; -- Argumento 4 Funcion 0.09801714
         when "00000101" => S <= "10001111"; -- Argumento 5 Funcion 0.12241068
         when "00000110" => S <= "10010010"; -- Argumento 6 Funcion 0.14673047
         when "00000111" => S <= "10010101"; -- Argumento 7 Funcion 0.17096189
         when "00001000" => S <= "10011000"; -- Argumento 8 Funcion 0.19509032
         when "00001001" => S <= "10011100"; -- Argumento 9 Funcion 0.21910124
         when "00001010" => S <= "10011111"; -- Argumento 10 Funcion 0.24298018
         when "00001011" => S <= "10100010"; -- Argumento 11 Funcion 0.26671276
         when "00001100" => S <= "10100101"; -- Argumento 12 Funcion 0.29028468
         when "00001101" => S <= "10101000"; -- Argumento 13 Funcion 0.31368174
         when "00001110" => S <= "10101011"; -- Argumento 14 Funcion 0.33688985
         when "00001111" => S <= "10101110"; -- Argumento 15 Funcion 0.35989504
         when "00010000" => S <= "10110000"; -- Argumento 16 Funcion 0.38268343
         when "00010001" => S <= "10110011"; -- Argumento 17 Funcion 0.40524131
         when "00010010" => S <= "10110110"; -- Argumento 18 Funcion 0.42755509
         when "00010011" => S <= "10111001"; -- Argumento 19 Funcion 0.44961133
         when "00010100" => S <= "10111100"; -- Argumento 20 Funcion 0.47139674
         when "00010101" => S <= "10111111"; -- Argumento 21 Funcion 0.49289819
         when "00010110" => S <= "11000001"; -- Argumento 22 Funcion 0.51410274
         when "00010111" => S <= "11000100"; -- Argumento 23 Funcion 0.53499762
         when "00011000" => S <= "11000111"; -- Argumento 24 Funcion 0.55557023
        when "00011001" => S <= "11001001"; -- Argumento 25 Funcion 0.57580819
        when "00011010" => S <= "11001100"; -- Argumento 26 Funcion 0.59569930
        when "00011011" => S <= "11001110"; -- Argumento 27 Funcion 0.61523159
        when "00011100" => S <= "11010001"; -- Argumento 28 Funcion 0.63439328
        when "00011101" => S <= "11010011"; -- Argumento 29 Funcion 0.65317284
        when "00011110" => S <= "11010101"; -- Argumento 30 Funcion 0.67155895
        when "00011111" => S <= "11011000"; -- Argumento 31 Funcion 0.68954054
        when "00100000" => S <= "11011010"; -- Argumento 32 Funcion 0.70710678
        when "00100001" => S <= "11011100"; -- Argumento 33 Funcion 0.72424708
        when "00100010" => S <= "11011110"; -- Argumento 34 Funcion 0.74095113
        when "00100011" => S <= "11100000"; -- Argumento 35 Funcion 0.75720885
        when "00100100" => S <= "11100010"; -- Argumento 36 Funcion 0.77301045
        when "00100101" => S <= "11100100"; -- Argumento 37 Funcion 0.78834643
        when "00100110" => S <= "11100110"; -- Argumento 38 Funcion 0.80320753
        when "00100111" => S <= "11101000"; -- Argumento 39 Funcion 0.81758481
        when "00101000" => S <= "11101010"; -- Argumento 40 Funcion 0.83146961
        when "00101001" => S <= "11101100"; -- Argumento 41 Funcion 0.84485357
        when "00101010" => S <= "11101101"; -- Argumento 42 Funcion 0.85772861
        when "00101011" => S <= "11101111"; -- Argumento 43 Funcion 0.87008699
        when "00101100" => S <= "11110000"; -- Argumento 44 Funcion 0.88192126
        when "00101101" => S <= "11110010"; -- Argumento 45 Funcion 0.89322430
        when "00101110" => S <= "11110011"; -- Argumento 46 Funcion 0.90398929
```

...

```
250             when "11101010" => S <= "00111110"; -- Argumento 234 Funcion -0.51410274
251             when "11101011" => S <= "01000000"; -- Argumento 235 Funcion -0.49289819
252             when "11101100" => S <= "01000011"; -- Argumento 236 Funcion -0.47139674
253             when "11101101" => S <= "01000110"; -- Argumento 237 Funcion -0.44961133
254             when "11101110" => S <= "01001001"; -- Argumento 238 Funcion -0.42755509
255             when "11101111" => S <= "01001100"; -- Argumento 239 Funcion -0.40524131
256             when "11110000" => S <= "01001111"; -- Argumento 240 Funcion -0.38268343
257             when "11110001" => S <= "01010001"; -- Argumento 241 Funcion -0.35989504
258             when "11110010" => S <= "01010100"; -- Argumento 242 Funcion -0.33688985
259             when "11110011" => S <= "01010111"; -- Argumento 243 Funcion -0.31368174
260             when "11110100" => S <= "01011010"; -- Argumento 244 Funcion -0.29028468
261             when "11110101" => S <= "01011101"; -- Argumento 245 Funcion -0.26671276
262             when "11110110" => S <= "01100000"; -- Argumento 246 Funcion -0.24298018
263             when "11110111" => S <= "01100011"; -- Argumento 247 Funcion -0.21910124
264             when "11111000" => S <= "01100111"; -- Argumento 248 Funcion -0.19509032
265             when "11111001" => S <= "01101010"; -- Argumento 249 Funcion -0.17096189
266             when "11111010" => S <= "01101101"; -- Argumento 250 Funcion -0.14673047
267             when "11111011" => S <= "01110000"; -- Argumento 251 Funcion -0.12241068
268             when "11111100" => S <= "01110011"; -- Argumento 252 Funcion -0.09801714
269             when "11111101" => S <= "01110110"; -- Argumento 253 Funcion -0.07356456
270             when "11111110" => S <= "01111001"; -- Argumento 254 Funcion -0.04906767
271             when "11111111" => S <= "01111100"; -- Argumento 255 Funcion -0.02454123
272             when others => null;
273         end case;
274     end process;
275  end Tabla;
276
```

C.4 Interfaz gráfica.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class Ondas extends JFrame {
        JButton botonOK_frecuencia, botonOK_amplitud;
        JTextField frecuencia, amplitud;
        JComboBox tipo_onda;
        String[] lista = {"Sinusoidal", "Cuadrada", "Triangular"};

        public Ondas() {
                setLayout(new FlowLayout());
                add(new JLabel("Frecuencia "));
                frecuencia = new JTextField(5);
                add(frecuencia);
                botonOK_frecuencia = new JButton("OK");
                add(botonOK_frecuencia);
                botonOK_frecuencia.addActionListener(new OyenteBoton1());
                add(new JLabel("          Amplitud "));
                amplitud = new JTextField(5);
                add(amplitud);
                botonOK_amplitud = new JButton("OK");
                add(botonOK_amplitud);
                botonOK_amplitud.addActionListener(new OyenteBoton2());
                add(new JLabel("                    Tipo de onda: "));
                tipo_onda = new JComboBox();
                for(int i=0; i<lista.length; i++)
                        tipo_onda.addItem(lista[i]);
                tipo_onda.addItemListener(new OyenteCombo());
                add(tipo_onda);
                setSize(540, 130);
                setVisible(true);
                setDefaultCloseOperation(EXIT_ON_CLOSE);
        }

        public static void main(String args[]) {
```

```
                Ondas ventana = new Ondas();
                }

        class OyenteBoton1 implements ActionListener {
                public void actionPerformed(ActionEvent e){
                        String valor_frecuencia_s = frecuencia.getText();
                        int valor_frecuencia = Integer.parseInt(valor_frecuencia_s);
                        if (valor_frecuencia >= 0) {
                                if (valor_frecuencia <= 50000) {
                                        int P = valor_frecuencia*86;
                                } else {
                JOptionPane.showMessageDialog(null,"Dato incorrecto","Error de entrada
",JOptionPane.ERROR_MESSAGE);
                                }
                        } else {
                JOptionPane.showMessageDialog(null,"Dato incorrecto","Error de entrada
",JOptionPane.ERROR_MESSAGE);
                        }
                        }
        }

        class OyenteBoton2 implements ActionListener {
                public void actionPerformed(ActionEvent e){
                        String valor_amplitud_s = amplitud.getText();
                        int valor_amplitud = Integer.parseInt(valor_amplitud_s);
                        }
        }

        class OyenteCombo implements ItemListener {
                public void itemStateChanged(ItemEvent e){
                        int ind = tipo_onda.getSelectedIndex();
                        }
        }
}
```

BIBLIOGRAFÍA

1.- International Association for the Study of Pain*: Unrelieved pain is a major global healthcare problem*.[http://www.iasp-pain.org/AM/Template.cfm?Section=Home&Template=/CM/ContentDisplay.cfm&ContentID=2908].

2.- https://www.eluniversal.com.mx/articulo/nacion/sociedad/2017/06/8/en-mexico-27-de-la-poblacion-padece-o-padecera-dolor-cronico.

3.- R. P. Walmsley , G. Letts , J. Vooys. (August 1, 1984). *A Comparison of Torque Generated by Knee Extension With a Maximal Voluntary Muscle Contraction vis-à-vis Electrical Stimulation*. Journal of Orthopaedic & Sports Physical Therapy, 6, 10-17.

4.- A. Chantraine, C. Gobelet, J. L. Ziltener. (1998). *Electrothérapie*. Kinésithérapie-Médecine physique-Réadaptation, 26-145-A-10, 22 pp.

5.- Ricardo Ávila Araya. (2000). *Corrientes Rusas. Introducción*. En Cátedra de Fisioterapia (16). Santiago, Chile: Universidad Pedro de Valdivia.

6.- Alex R. Ward, Nataliya Shkuratova. (2002). *Russian electrical stimulation: the early experiments*. Physical Therapy, 82, 1019-30.

7.- Álvaro Cortes Mánica. (2008). Tesis: *Diseño de un dispositivo generador de corrientes interferenciales para el tratamiento de patologías de origen muscular y circulatorio*. Mérida, Yucatán, México: Instituto Tecnológico de Mérida.

8.- Luis Bernal. (2008). *Baja, media y alta frecuencia*. 2 jun 2020, de luisbernal.com Sitio web: http://www.sld.cu/galerias/pdf/sitios/rehabilitacion/tecnicas_electroterapicas_con_corrientes_de_baja,_media_y_alta_frecuencia.pdf

9.- Aleksander Malinowski and Hao Yu. (May, 2011). *Comparison of Embedded System Design for Industrial Applications*. IEEE Transaction on Industrial Informatics, 7, No. 2, 244-254.

10.- Peter Bishop . (2009). *A tradeoff between microcontroller, DSP, FPGA and ASIC technologies*. Jun 2020, de EE|Times Sitio web: https://www.eetimes.com/a-tradeoff-between-microcontroller-dsp-fpga-and-asic-technologies/#

11.- Arrow. (Oct. 2018). *FPGA vs CPU vs GPU vs Microcontroller: How Do They Fit into the Processing Jigsaw Puzzle?*. Jun 2020, de Arrow Sitio web: https://www.arrow.com/en/research-and-events/articles/fpga-vs-cpu-vs-gpu-vs-microcontroller

12.- Intel, *Pentium Processor 75/90/100/120/133/150/166/200 - Datasheet*', Sitio web: http://www.dexsilicium.com/Intel_Pentium.pdf

13.- https://ark.intel.com/content/www/es/es/ark/products/120489/intel-xeon-gold-6148-processor-27-5m-cache-2-40-ghz.html

14.- Microchip. (2005). *PIC10F200/202/204/206 Data Sheet*. Jun 2020, de Microchip Technology Inc. Sitio web: http://ww1.microchip.com/downloads/en/DeviceDoc/40001239F.pdf

15.- Texas Instruments. (2005). *TMS320C6211, TMS320C6211B Fixed-Point Digital Signal Processors*. Jun 2020, de Texas Instruments Sitio web: https://www.ti.com/lit/ds/sprs073l/sprs073l.pdf

16.- Debbie Greenstreet, John Smrstik. (May 2017). *Voice as the user interface – a new era in speech processing*. jun 2020, de Texas Instruments Sitio web: https://www.ti.com/lit/wp/slyy116/slyy116.pdf

17.- http://www.xilinx.com

18.- Texas Instruments. (2002). *Choosing the Right Architecture for Real-Time Signal Processing Designs*. jun 2020, de Texas Instruments Inc. Sitio web: https://www.ti.com/lit/wp/spra879/spra879.pdf

19.- Altera. (2007). *FPGA vs. DSP Design Reliability and Maintenance*. jun 2020, de Intel-FPGA Sitio web: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/wp/wp-01023.pdf

20.- Oshana, Robert. (2012). *Overview of Embedded Systems Development Life Cycle Using DSP*. En DSP for Embedded and Real-Time Systems (35-47). Elsevier: Newnes.

21.- Abdala Gustavo Hugo. (2017). ECAM Manual de Instrucciones Estetic RCT. jun 2020, de ACCME Equipamiento Médico Sitio web: https://accme.com.ar/

22.- Fisaude. (Nov 2005). Manual de Instrucciones Electroestimulador muscular analgésico de dos canales Bimodal TENS. jun 2020, de Fisaude Sitio web: https://www.fisaude.pt/files/Ficha%20Tecnica%20Tens%20Biomodal.pdf

23.- https://la.mathworks.com

24.- https://www.intel.la/content/www/xl/es/software/programmable/overview.html