

Trabajo Fin de Grado

"Integración de control, medida y ensayos de un electroporador para tratamientos quirúrgicos".

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación.

Autor/es

María Moreno Sau

Director/es

Carlos Bernal Ruiz
José Miguel Burdío Pinilla

Escuela de Ingeniería y Arquitectura
Universidad de Zaragoza

Septiembre 2015

Resumen

El presente trabajo está enmarcado en una línea de colaboración de la Universidad de Zaragoza con un grupo multidisciplinar (formado por cirujanos, veterinarios e ingenieros) de otras Universidades y centros de investigación.

Se colaborará en la investigación del diseño de un sistema electrónico para la aplicación de tratamientos con niveles de tensión elevados (en torno a 15kV) que permitan verificar de manera experimental los rangos de aplicación de la electroporación sin que la barrera tecnológica de campo eléctrico, 2500V/cm, o mantenimiento del pulso, 100 μ s, sean limitantes.

La electroporación es una técnica que consiste en la exposición celular a campos eléctricos de alta intensidad, favoreciendo de este modo el aumento de la permeabilidad de sus membranas y permitiendo la circulación de iones y macromoléculas a través de esta. Este aumento de la permeabilidad está vinculado a la afectación de todo el contorno de la membrana celular.

Hay dos tipos de electroporación, electroporación reversible (EPR) e irreversible (EPI). En la segunda, la afección en la membrana no es reversible y por tanto la célula no es capaz de recuperar el equilibrio y se desencadena la lisis celular, muerte natural. En la persecución de la muerte celular, la EPI es el método de electroporación más directo y sus propiedades pueden usarse en tratamientos médicos. La principal ventaja de la electroporación, es que induce la muerte celular sin producir calentamiento del tejido circundante, evitando así sus efectos adversos en el resto de tejido sano.

Se tendrá como propósito 3 hitos principales. Se comenzará por el estudio y análisis del pulso de electroporación partiendo del modelo eléctrico generador-tejido, con el objetivo de cuantificar, para cada forma de onda de pulso, cuál es su ancho de banda efectivo. La siguiente meta a superar, será el diseño e implementación de una topología de acoplador/atenuador capaz de capturar la forma de onda procedente del tratamiento aplicado al tejido. Dicha topología, deberá conseguir baja distorsión de pulso, atenuación planteada y cumplir BW adecuado independientemente de la carga del osciloscopio o de las tolerancias de los parámetros. Por último, en este Trabajo Fin de Grado se desarrollará un proyecto software de control básico que permita al personal cualificado, el dominio de los parámetros de la forma de onda requerida, así como de su creación y aplicación en tiempo real.

Como objetivo final se plantea, la verificación y puesta en marcha de los prototipos a desarrollar.

Índice de contenidos

1	Introducción y motivación	6
1.1.-	Contexto	6
1.2.-	Aplicación	7
1.3.-	Sistema y Diseño del equipo.....	7
1.3.1.-	Sistema electroporador.....	7
1.3.2.-	Diseño del equipo completo.....	9
1.4.-	Problemas y tareas no resueltos	10
1.5.-	Objetivos y Alcance	11
2	Estudio del pulso de Electroporación	14
2.1.-	Pulso BURST con electroporación	14
2.2.-	Pulso SINGLE con electroporación.....	17
2.3.-	Pulso BURST y SINGLE sin electroporación.....	19
3	Acoplador – Capturador de onda completa	22
3.1.-	Topología propuesta.....	22
3.2.-	Simulación.....	28
3.2.1.-	Topología 5 niveles.	29
3.2.2.-	Topología 11 niveles.....	30
3.3.-	Diseño y montaje sobre PCB	32
3.4.-	Testeo y verificación.....	35
3.4.1.-	Análisis frecuencial en pequeña señal	35
3.4.2.-	Análisis en gran señal	38
3.4.3.-	Análisis en DC	39
3.4.3.-	Caracterización completa del sistema atenuador.....	39
3.4.4.-	Mejoras y modificaciones	41
4	Creación de Software de control	44
4.1.-	Objetivo	44
4.2.-	Ingeniería de Software: Análisis.....	44
4.3.-	Ingeniería de Software: Diseño y Código	45
4.3.1.-	Firmware.....	48
4.3.2.-	Protocolo de comunicación	52
4.3.3.-	Comunicación Bluetooth.....	53
4.3.4.-	App Android.....	55
4.4.-	Ingeniería de Software: Pruebas y Mantenimiento	59
5	Conclusiones	65

ANEXO 1 : Código Matlab para el estudio del pulso de electroporación.....	67
ANEXO 2 : Selección de componentes para PCB del acoplador.	73
ANEXO 3 : Código Arduino para software de control.	74
ANEXO 4 : APP INVENTOR 2.	78
ANEXO 5 : Variables, componentes y bloques de App Android.....	80
ANEXO 6 : REFERENCIAS y BIBLIOGRAFÍA.....	83

1 Introducción y motivación

1.1.- Contexto

El grupo de investigación de Electrónica de Potencia y Microelectrónica (GEPM) dentro del instituto de investigación I3A de la Universidad de Zaragoza, en colaboración con la Universidad Pompeu Fabra, el Hospital del Mar de Barcelona y otros socios de investigación, se encuentra desarrollando un sistema experimental para el análisis de las técnicas de electroporación irreversible aplicadas al tratamiento de tumores hepáticos.

A continuación se presenta una visión general de los aspectos principales de la Electroporación (EP), así como sus fundamentos científicos. Lo cual está orientado a la construcción de un prototipo de experimentación con los niveles de tensión y forma de onda requeridos.

Electroporar [1] consiste en exponer las células a campos eléctricos de alta intensidad. Dicho campo induce el aumento de la permeabilidad de la membrana celular favoreciendo la circulación de iones y macromoléculas a través de ella. El aumento de la permeabilidad está vinculado a la generación de pasos permeables a lo largo de la membrana, este efecto es representado en la Figura 1.

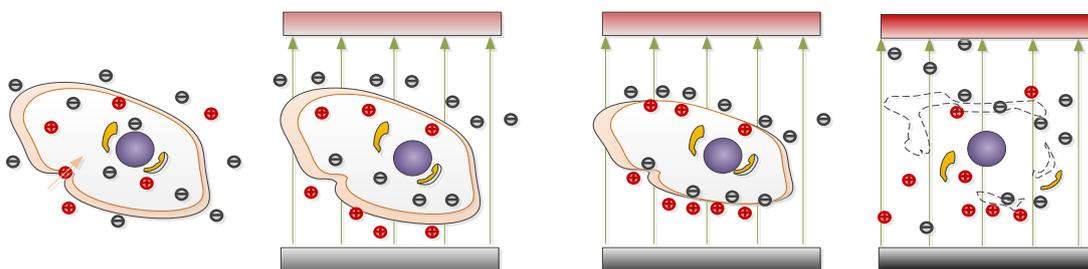


Figura 1 Electroporación de una célula

Una vez suspendido el campo eléctrico, los mecanismos de auto-reparación permiten restablecer los poros generados recobrándose así la permeabilidad selectiva que favorece la homeostasia de la célula. Si esto se produce, podemos hablar de lo que se denomina electroporación reversible (EPR). En cambio, si la estimulación de la membrana ha sido excesiva puede haber sufrido una afectación no reversible, a este efecto se le denomina electroporación irreversible (EPI). En condiciones extremas de deterioro de la membrana, la célula no es capaz de recuperar el equilibrio y se desencadena la lisis celular, una muerte de forma natural.

1.2.- Aplicación

Las características de la EPI son clave en algunos tipos de intervenciones, sobre todo por su carácter no invasivo y la baja generación de secuelas post-tratamiento, frente a las graves repercusiones que puede producir un tratamiento con agentes citotóxicos o ionizantes. Por tanto, el desarrollo y control de esta técnica puede suponer un gran hito en el tratamiento oncológico de cara a la duración y efectividad en los tratamientos.

La principal ventaja que presenta la EPI es que provoca la muerte celular instantánea sin producir el calentamiento del tejido tratado ni del circundante.

Una desventaja es que durante la aplicación de la EPI puede producirse un alto grado de contracción muscular en zonas cercanas o alejadas, este efecto puede ser mitigado con la administración de relajantes musculares. Además, algunas medidas preventivas respecto a la sincronización del tratamiento con la contracción cardiovascular se deben tener en cuenta para evitar inducir contracciones involuntarias del miocardio. Por último, cabe destacar que la fácil caracterización del tejido y del tipo de pulso permite desarrollar planificaciones de tratamiento más sencillas que las requeridas en otras técnicas, como por ejemplo la radioterapia.

1.3.- Sistema y Diseño del equipo

1.3.1.- Sistema electroporador

Para poder planificar y estimar la efectividad de los tratamientos es necesario conocer las características claves de la evolución del tejido durante la macroporalización. Resulta de utilidad disponer pues de un modelo equivalente eléctrico sobre el que extrapolarlas. Disponer de este modelo permite estimar el rango de variabilidad de la carga equivalente durante el tratamiento y la manera en que éste puede influir.

Se va a utilizar un modelo de tejido (carga) desarrollado por miembros del equipo de investigación y presentado en [2]. En él, se tiene en cuenta que cada porción infinitesimal puede modelarse como resistencias (R_e y R_i), dado que su contenido está compuesto casi en su totalidad por iones. Del mismo modo, para caracterizar la membrana, se aproxima a una serie de capacidades infinitesimales en serie y paralelo, dado lugar a la capacidad equivalente de la membrana (C_m). El circuito que vemos en la Figura 2 equivale al visto desde los terminales de los electrodos.

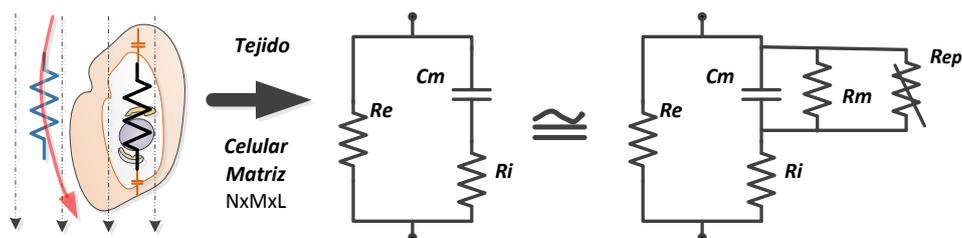


Figura 2 Circuito eléctrico equivalente del tejido

Se caracteriza en él también, los efectos de la electroporación sobre el carácter dieléctrico de la membrana. El efecto de conductividad residual que se manifiesta con la EP se representa mediante una resistencia variable (R_{ep}) y surge el deterioro del carácter dieléctrico. Una vez conseguida la permeabilidad en la membrana, se puede caracterizar como una resistencia R_m , aproximable a un cortocircuito.

Existen además, dos parámetros adicionales que afectan a la impedancia equivalente del sistema: la frecuencia de estimulación y el estado de electroporación. Para altas frecuencias (10kHz a 1MHz) la corriente eléctrica es capaz de atravesar la membrana de las células, reduciendo la impedancia del conjunto, como vemos en la Figura 3.

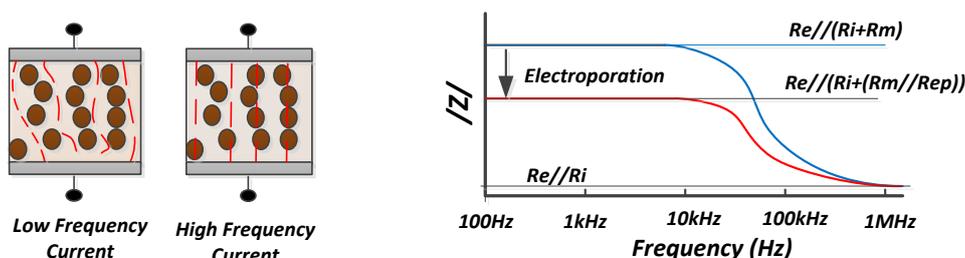


Figura 3 Afectación de la impedancia con la frecuencia y la electroporación

Esto implica que aunque se usen menores niveles de tensión a los considerados en la EPI, si se mantiene un tiempo de exposición suficiente, puede llegar a inducir la muerte necrótica o apoptótica [3].

El grupo de investigación intenta desarrollar un prototipo que permita alcanzar la mayor tasa de verticalidad del pulso intentando favorecer a su vez el mantenimiento de la tensión de salida con las mayores anchuras posibles de pulso.

Realizar la electroporación de un hígado completo resulta inviable para los sistemas actuales, dados los grandes niveles de tensión y corriente requeridos para cargar las grandes capacidades asociadas al tejido. Por ello, el tratamiento se aplica por segmentos. El modelo de carga de cada segmento a electroporar es aproximadamente un cilindro de tejido blando (hígado) definido por la superficie de

contacto de dos electrodos circulares (5 cm de diámetro) con una separación entre ambos (6 cm). El nivel de campo eléctrico debe establecerse en un rango superior a 2000 V/cm.

A través de varios estudios se ha descubierto que, mediante la perfusión de suero salino en los tejidos sanos, es posible afectar regiones de tejido con tensiones menores a las originales. Se ha constatado además que los tejidos sanos pasan a conducir con mayor facilidad, reduciéndose el potencial eléctrico especialmente en la zona afectada por irrigación y estableciéndose un mayor campo eléctrico en la zona de baja conductividad correspondiente al tumor. Así se puede conseguir una electroporación selectiva, reduciendo el deterioro del tejido sano y se va aumentando el rendimiento y especificad del tratamiento.

Estudios realizados por el grupo de desarrollo establecen que una intensidad de campo de 1900-2000 V/cm en tejido hipersalino puede ser suficiente para la macroelectropolación selectiva. Sin embargo, uno de los integrantes del equipo ha intentado obtener una intensidad de campo de 2500 V/cm para nuevas pruebas. Dado que la longitud del cilindro de aplicación propuesto es de 6 cm, el equipo desarrollado es capaz de administrar pulsos bipolares de 15 kV. La herramienta, permite proporcionar pulsos variables entre 10 y 100 μ s, bipolares, ciclo de trabajo simétricos y secuencia repetitiva cada segundo.

La impedancia equivalente en el peor de los casos, correspondiente con la poración, es de $R_e // R_i \cong 50 \Omega$, y las tensiones rondan los kV. Estas magnitudes implican corrientes de centenares de amperios (para decenas de kV) que se traducen en potencias instantáneas del orden de algunas unidades de MW.

1.3.2.- Diseño del equipo completo

Con el fin de poder concretar la eficacia del procedimiento, no sólo con la amplitud máxima alcanzada sino también con el mantenimiento de la misma y repetitividad, es necesario disponer de un equipo que permita variar amplitud, frecuencia y ciclo de trabajo con modulación bipolares.

La topología seleccionada es de tipo indirecto denominada como HVPS (*High Voltage Pulse Supply*), y la señal aplicada no es generada directamente sino que se genera a menor voltaje y posteriormente se eleva y acopla magnéticamente, mediante un transformador como muestra la Figura 4.

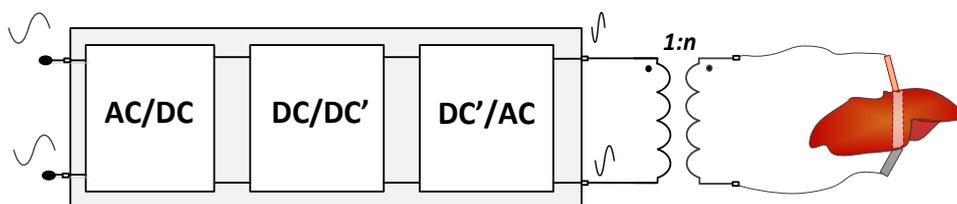


Figura 4 Esquema de la estructura de electrónica de potencia del sistema EPI

La topología que resuelve el acoplamiento magnético de la carga está estudiada por el grupo de investigación. Con el objetivo de obtener 15kV en la carga, asegurando los 2500 V/cm en cargas de 6 cm de espesor, con una tensión de primario cercana a 600V.

Finalmente, tras su desarrollo, el circuito eléctrico aproximado de la totalidad del sistema junto a la carga biológica quedaría del modo representado en Figura 5.

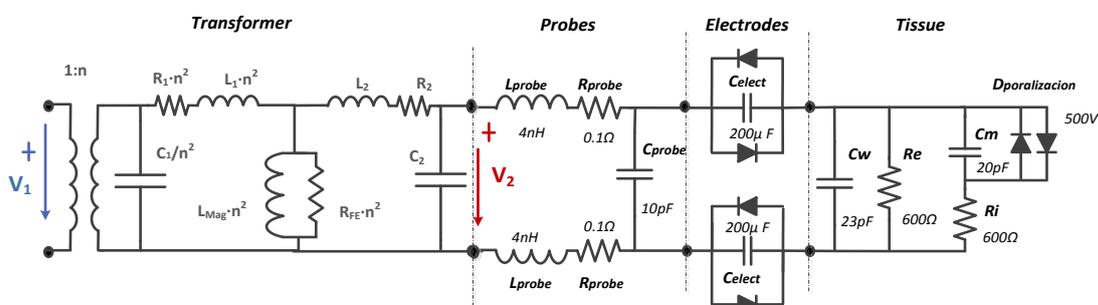


Figura 5 Circuito eléctrico aproximado de la totalidad del sistema y carga biológica

1.4.- Problemas y tareas no resueltos

En este punto cabe plantear las tareas no resueltas en el sistema para el tratamiento de electroporación irreversible.

Veamos una visión general del proyecto en el siguiente diagrama de bloques.

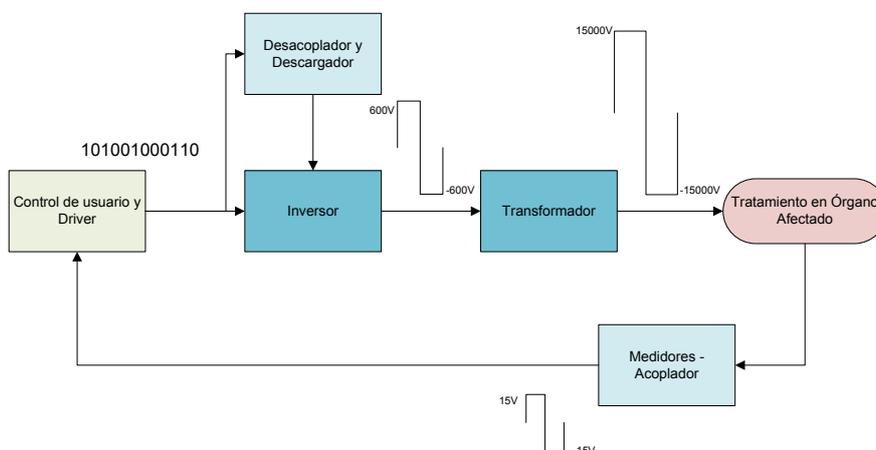


Figura 6 Diagrama de bloques sistema electroporador

Como vemos, es necesario abarcar partes del mismo que todavía no han sido evolucionadas. Este trabajo fin de grado realiza aportaciones en dos aspectos: la medida del pulso de alta tensión y el control electrónico de pulso.

Por una parte, será necesario conseguir una topología de acoplador, que permita capturar la onda procedente del resultado del tratamiento. Para ello, dado que se aplican altos valores de tensión (decenas de kV) en el paciente, se debe obtener la forma de onda consiguiendo a su vez la atenuación necesaria para que un dispositivo electrónico consiga procesar la misma. Esto permitirá su posterior estudio, y la obtención de conclusiones y mejoras en el tratamiento de EPI.

Por otro lado, resulta imprescindible abarcar el control y la generación de los pulsos. De forma que el usuario, en nuestro caso el investigador médico o veterinario especialista en el tratamiento, tenga control sencillo sobre los parámetros del proceso. Esto nos impulsa al desarrollo de software que sintetice las formas de onda a la entrada del driver del sistema, y permita ajustar sus características. Así mismo se necesitará el desarrollo una aplicación para dispositivos móviles, de forma que los pulsos puedan controlarse sin contacto galvánico, la comunicación necesaria con el software implementado y el análisis de las formas de onda generadas.

1.5.- Objetivos y Alcance

En el apartado anterior, se han especificado tareas pendientes de resolver en el sistema dedicado al tratamiento de tumores por EPI. De modo, que esos mismos cometidos serán gran parte de la ocupación del presente trabajo fin de grado. A continuación se detallan cada uno de los propósitos que se desarrollarán.

- Estudio de la electroporación celular y macroelectroporación, así como el sistema desarrollado hoy en día para llevar a cabo el tratamiento. Dicho hito, con el fin de conocer de forma clara el contexto en el que se desarrollarán las posteriores tareas a tratar.
- Estudio y análisis del pulso de electroporación. Partiendo del modelo eléctrico del conjunto generador - tejido es posible analizar el tipo de pulso que se va a extraer en el tejido electroporado. Para ello se tendrá en cuenta que se van a manejar diferentes modulaciones (pulso único repetido, trenes de pulsos, bipolar o unipolar). El objetivo del análisis trata de cuantificar, para cada forma de onda de pulso, cuál es su ancho de banda efectivo.
- Diseño e implementación de una topología de acoplador capaz de capturar la forma de onda procedente del tratamiento aplicado en el tejido. Dicha etapa, deberá mantener una baja distorsión de pulso y atenuar 1/1000 veces el mismo. Conociendo el BW que cumplir, obtenido en la tarea anterior, se intenta cumplir el ancho de banda que lo supere, proponiendo conseguir que la etapa finalmente deje de tener dependencia frecuencial.

- Desarrollar un proyecto software de control básico que permita, al personal cualificado, el dominio de los parámetros de la forma de onda requerida, así como de su creación y aplicación en tiempo real. Esta tarea, abarca además la implementación de una aplicación dirigida a su uso en un dispositivo móvil, la comunicación imprescindible con el software desarrollado y la verificación final.
- Obtención de conclusiones sobre la viabilidad de las tareas desarrolladas en la aplicación real, basadas en los resultados de los prototipos desarrollados. Y propuesta de mejoras y alternativas.

2 Estudio del pulso de Electroporación

Partiendo del modelo eléctrico del conjunto generador - tejido es posible analizar el tipo de pulso que se va a extraer en el tejido electroporado. Para ello se tendrá en cuenta que se van a manejar diferentes modulaciones (pulso único repetido, trenes de pulsos, bipolar o unipolar).

El objetivo del análisis que se va a presentar a continuación es cuantificar, para cada forma de onda de pulso, cuál es su ancho de banda efectivo. Para poder concretar los requisitos de BW en toda la cadena de análisis y tratamiento, se tomará en consideración el ancho de banda que contiene una determinada potencia, de forma paralela a lo que se hace en canales modulados de espectro no contenido [4].

En consecuencia se ha procedido a simular en el entorno PLECS (Matlab-Simulink) el conjunto generador - pulso, donde diferenciaremos entre cuatro posibles situaciones. El uso del entorno PLECS permite extraer las respuestas transitorias a los cambios de estado de un convertidor de forma más eficiente que la simulación clásica (p.e. Spice) La forma de onda aplicada en el tejido afectado por el tumor, será nuestro objetivo a estudiar.

Consideramos células del órgano electroporadas cuando se consiguen alcanzar los $\pm 15000V$ de tensión a la salida del transformador. Si el sistema no alcanza dicho voltaje, el tejido afectado no conseguirá electroporación de células afectadas.

2.1.- Pulso BURST con electroporación

Se estudia en este punto, la caracterización de secuencias de señal de 1 segundo de periodo, las cuales estarán formadas por trenes de pulsos a 50000 Hz (20us de tiempo de periodo) anexionados a un tiempo en reposo hasta complementar 1 segundo.

El modelo simulado en esta situación será mostrado en la Figura 7.

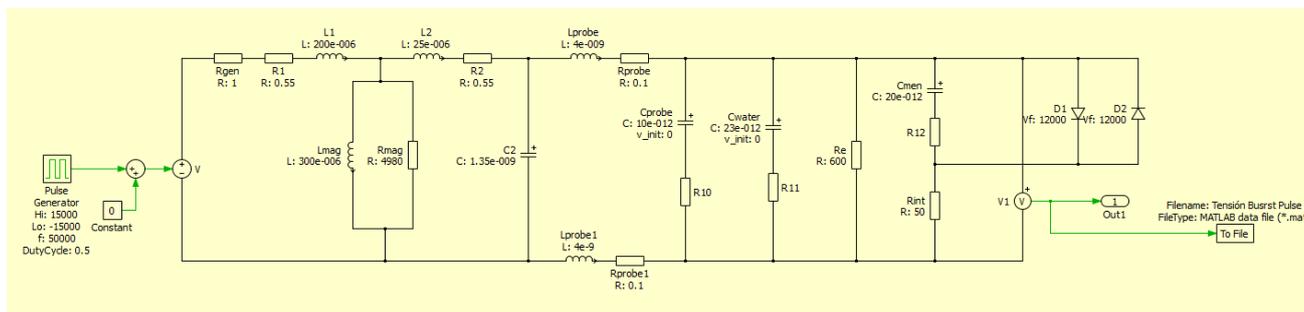


Figura 7 Modelo eléctrico simulado del sistema con carga biológica en pulso BURST

Se agrega a la entrada una señal de onda cuadrada de amplitud $\pm 15000V$. Las especificaciones ya expuestas anteriormente, dictan que la frecuencia de trabajo de la señal se fijará en 50000 Hz, lo que supone periodos de pulso de 20 μs . En dichos pulsos, la mitad del periodo tendrá amplitud positiva, y la segunda, negativa (duty de 0.5).

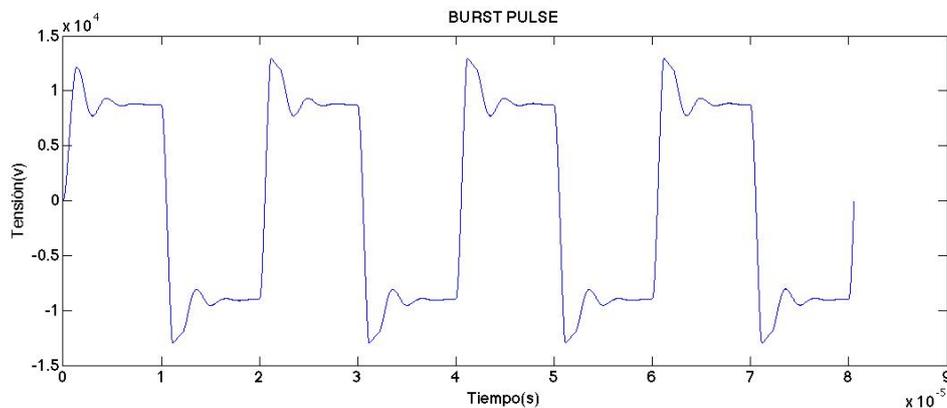


Figura 8 Forma de onda que llega al tejido haciendo aplicación de pulso BURST

Una vez visualizada la forma de onda que llega a la carga del paciente, que incluye tanto la onda cuadrada como las respuestas a los escalones de pulso, respuesta del circuito no lineal, deberemos importar los datos. En la imagen superior, vemos los cuatro primeros periodos que llegarán al tejido.

El primero de ellos, tiene ciertas diferencias en forma con respecto a los posteriores, que conforman el régimen permanente de la señal. Sin embargo, esa pequeña diferencia, no afecta a nuestro estudio, y por lo tanto, también se toman esos datos para el estudio energético de la señal.

La forma de onda de la señal que estudiamos, tiene características que nos dan información sobre lo que se produce en el sistema y eficiencia del tratamiento. En esta ocasión se percibe:

- Sobreoscilación en el techo del pulso.
- Caída y elevación del pulso, no totalmente vertical.

Estas características de onda, nos impulsan a decantarnos por un modelo numérico del estudio de energía. Mientras que si se tratasen de pulsos cuadrados tendría más sentido el estudio analítico de los armónicos en frecuencia de la señal.

Exportamos los datos procedentes de los pulsos que llegan a la carga en un archivo .mat. Este archivo es importado en Matlab y ejecutado en el mismo entorno.

El propósito se basa en conocer la energía que contiene la señal en función del ancho de banda seleccionado, haciendo un barrido de frecuencias. Consideramos como porcentaje aceptable del orden del 80% o superior.

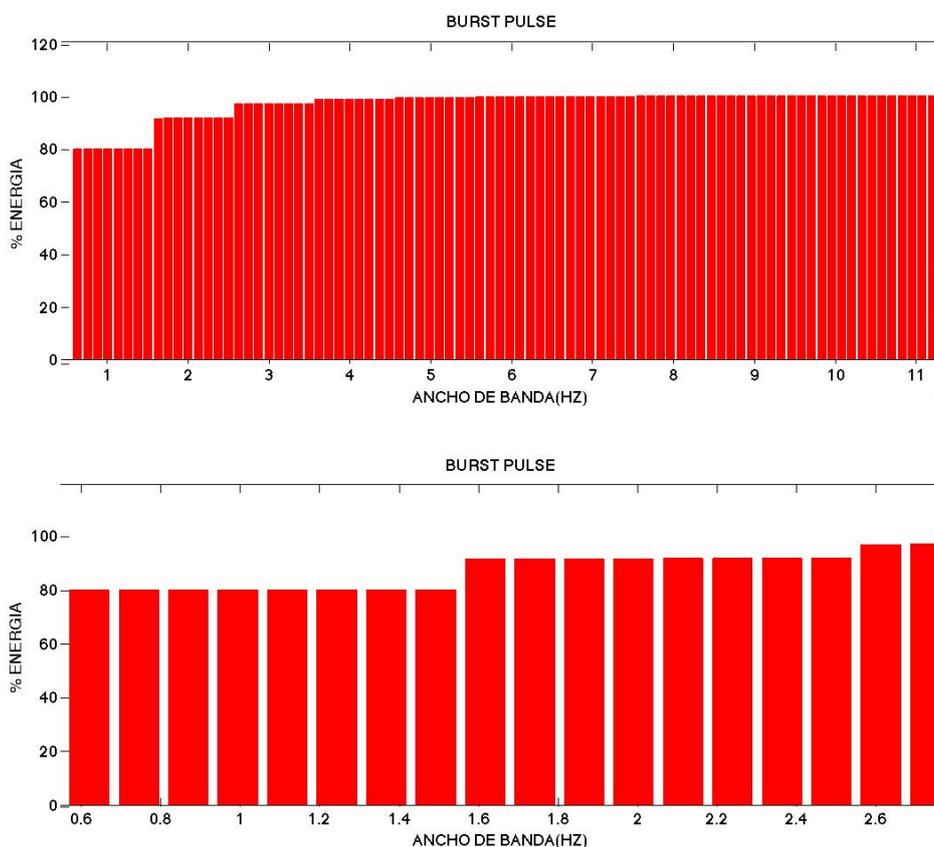


Figura 9 Zoom Energía contenida en función de la frecuencia de señal con pulso BURST

El código desarrollado que obtiene el estudio energético en función del ancho de banda, se especifica detalladamente en el ANEXO 1.

Como resultado de nuestro estudio, la Figura 9 muestra que para alcanzar el 80% de la energía necesitaremos usar 0.15 MHz de BW. Y para captar el 90% de la energía de la señal, será necesario un ancho de banda de 0.16 MHz.

2.2.- Pulso SINGLE con electroporación

En esta ocasión se investiga una circunstancia diferente a la anterior expuesta. Los valores de la señal serán análogos a los ya nombrados en cuanto a amplitud y frecuencia de cada pulso de la ráfaga. Sin embargo, únicamente es implementado en cada secuencia de 1 segundo, uno de esos pulsos de amplitud $\pm 15000V$.

En cuanto al modelo circuital que se desarrolla para su obtención es idéntico al representado en la Figura 7. Únicamente necesitaremos modificar la forma de onda de entrada al sistema.

Trabajando de nuevo a 50000 Hz, multiplicamos la señal de entrada por un pulso que tendrá amplitud unidad en el primer periodo, es decir, durante 20 μs , y el resto del tiempo, hasta pasar un segundo, el pulso permanecerá en valor 0.

Esto implica que el pulso por el que multiplicamos la señal cuadrada tendrá frecuencia 1Hz (periodos de 1 segundo) en el que el DutyCycle es 0.00002. Es decir, 1 de cada 50000 tiempos, tiene amplitud uno.

La forma de onda, simulada e importada en Matlab, que obtenemos es plasmada en Figura 10.

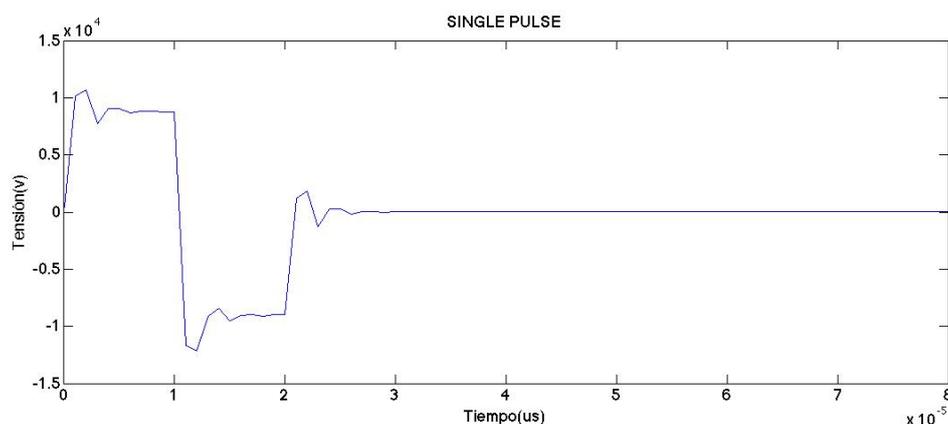


Figura 10 Forma de onda que llega al tejido haciendo aplicación de pulso SINGLE.

De igual forma que importábamos en Matlab los datos en el anterior caso, lo hacemos de esta señal. Con los valores procedentes de la señal, se realiza el estudio energético de la misma, en función del ancho de banda. La programación para su obtención se especifica detalladamente en el ANEXO 1.

Si representamos los porcentajes de energía contenida en función del ancho de banda seleccionado, vemos la siguiente Figura 11.

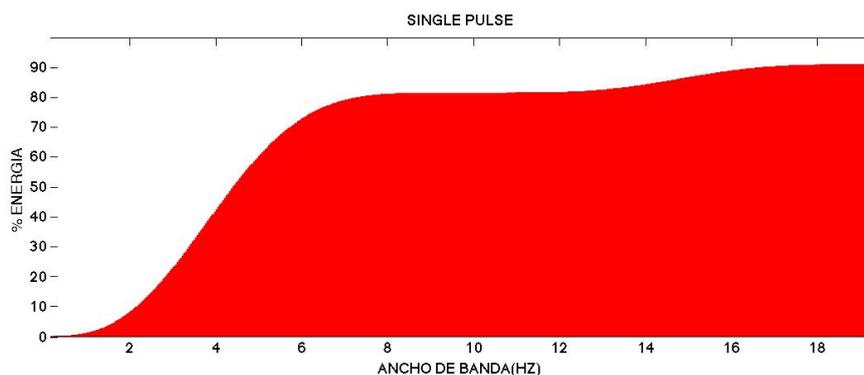


Figura 11 Energía contenida en función de la frecuencia de la señal con pulso SINGLE



Figura 12 Zoom 1 Energía contenida en función de la frecuencia se señal con pulso SINGLE

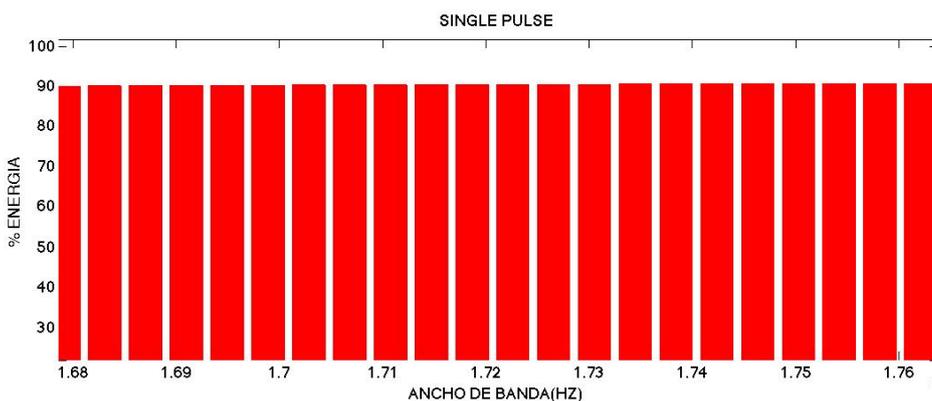


Figura 13 Zoom 2 Energía contenida en función de la frecuencia se señal con pulso SINGLE

Como señalábamos anteriormente, tomaremos como valor correcto, la captación de mínimo un 80% de la energía, lo que nos impondrá restricciones de BW a seleccionar. En el caso de pulso single, alcanzaremos el 80% de la energía seleccionando 0.073 MHz de BW, mientras que el 90% se alcanzará tomando 0.17 MHz de BW.

Respecto a los casos, en los que se produce electroporación, podemos concluir que tomando 0.2 MHz de ancho de banda, cumpliríamos la calidad de señal deseada en cuanto a energía captada.

2.3.- Pulso BURST y SINGLE sin electroporación

Cuando no ha sido posible alcanzar el voltaje deseado, y por tanto la onda cuadrada, no llega a los $\pm 15000V$, se considera que el tratamiento de electroporación no ha sido efectivo, y la señal en la carga tiene diferente comportamiento.

Esto tendrá como consecuencia, una señal muy diferente aplicada en el tejido afectado, lo que nos hace estudiar de nuevo la energía contenida en función del ancho de banda seleccionado en estas nuevas situaciones.

La figura 14 muestra cuatro periodos de pulso, representando la secuencia BURST.

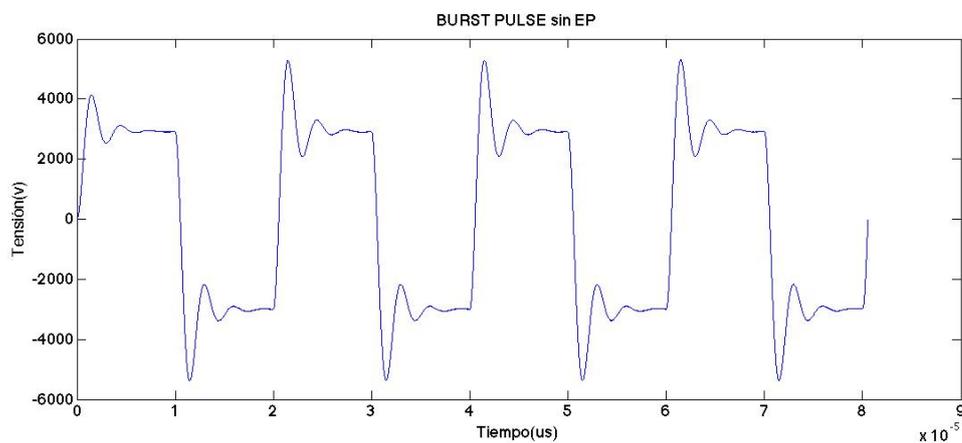


Figura 14 Forma de onda que llega al tejido aplicando pulso BURST sin electroporación

A continuación en la Figura 15, se representa un solo pulso en la secuencia de 1 segundo, lo que se interpreta como pulso SINGLE.

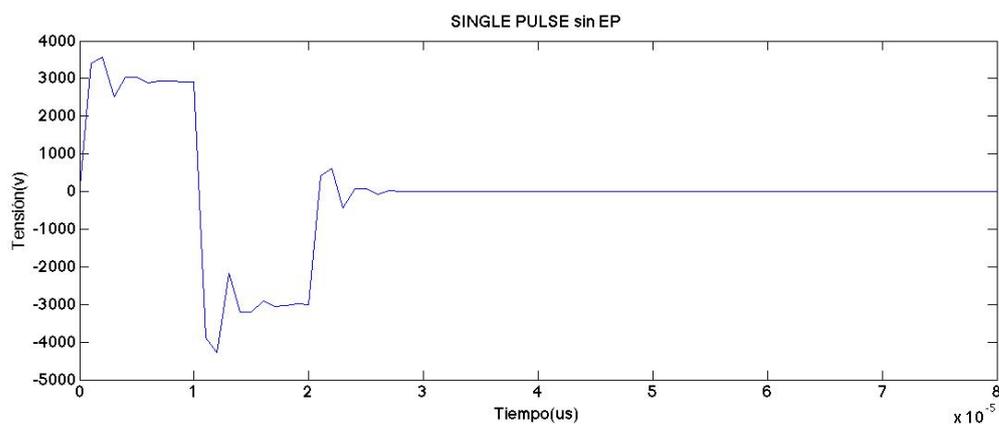


Figura 15 Forma de onda que llega al tejido aplicando pulso SINGLE sin electroporación

Observamos una sobreoscilación mayor en el techo de los pulsos, así como la no verticalidad de los mismos, que vuelve a impulsar el estudio energético en potencia de la señal.

En el estudio energético en función de ancho de banda en los dos tipos de pulso (burst y single), en situación de no electroporación, observamos que se producirán modificaciones en los BW escogido anteriormente. Veamos en las siguientes figuras (Figura 16 y 17) el porcentaje de energía contenida en función del ancho de banda.

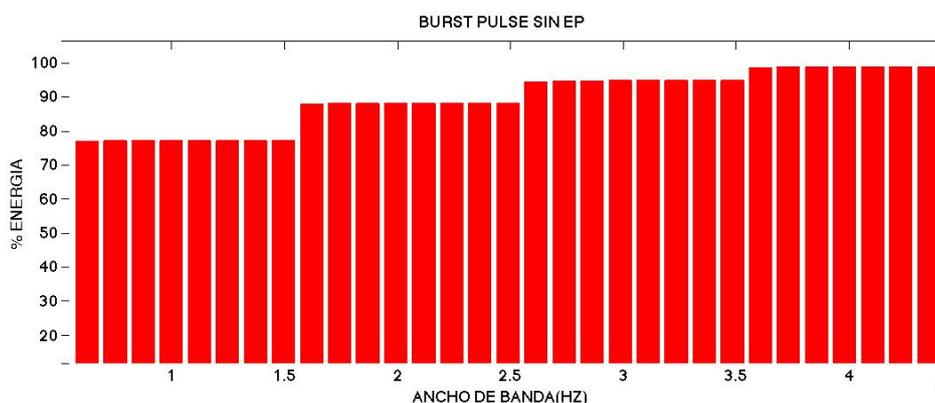


Figura 16 Energía contenida en función de la frecuencia de señal con pulso BURST sin electroporación

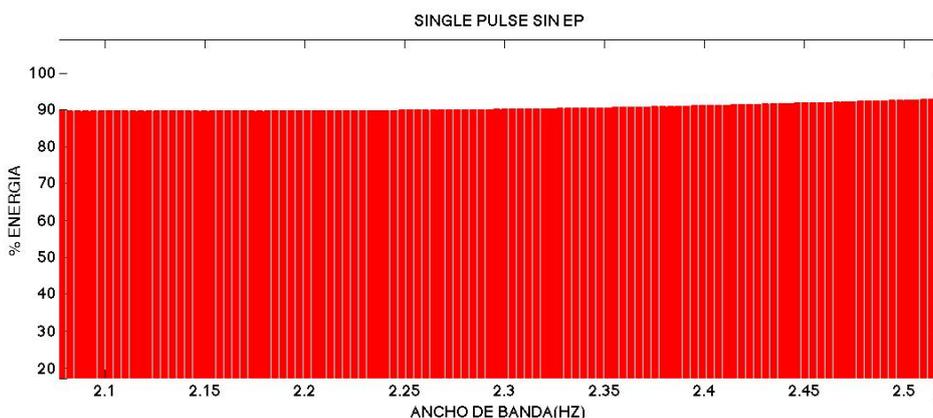


Figura 17 Energía contenida en función de la frecuencia de señal con pulso SINGLE sin electroporación

Para conseguir alcanzar un 90% de la energía, los anchos de banda mínimos serán 0,25 MHz y 0, 23 MHz en pulso Burst y Single respectivamente.

Estos últimos cálculos, nos permiten llegar a la conclusión de que coger BW de 0,26 MHz permitiría captar de forma adecuada la señal.

3 Acoplador – Capturador de onda completa

Una vez determinado el ancho de banda que cumplir en la captación de la señal que se aplica al tejido, buscamos un sistema capaz de capturar y atenuar dicha onda manteniendo una baja distorsión de pulso. Se trata del diseño de un circuito que permita atenuar 1/1000 veces el pulso, que admita además considerar los efectos de parásitos de la carga del osciloscopio.

Como objetivo, encontrar la topología de acoplador con atenuación 1000:1 que cumpla el BW especificado. Para ello, se intenta alcanzar un BW infinito, que nos permitirá trabajar de forma independiente a la frecuencia.

3.1.- Topología propuesta

Partiremos del atenuador de voltaje ideal formado por dos componentes resistivos de la forma mostrada en la Figura 18.

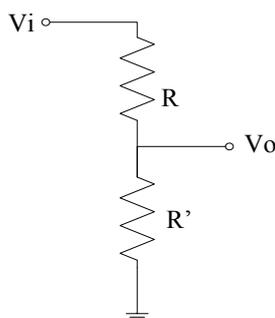


Figura 18 Divisor de voltaje resistivo

La relación entre V_o y V_i , será:

$$V_o = a * V_i$$

Siendo a , la atenuación producida por el divisor resistivo, y dependerá únicamente del valor de las resistencias:

$$a = \frac{R'}{R' + R}$$

En este primer caso, el circuito atenuador sería independiente de la señal de entrada, si no fuese por la presencia inevitable de capacitancia de entrada en la

etapa que sigue. En nuestro caso estimaremos 25 pF, a la entrada del osciloscopio, producidos por el circuito equivalente de entrada del osciloscopio. De este modo, el circuito que realmente tendríamos sería el mostrado en la Figura 19.

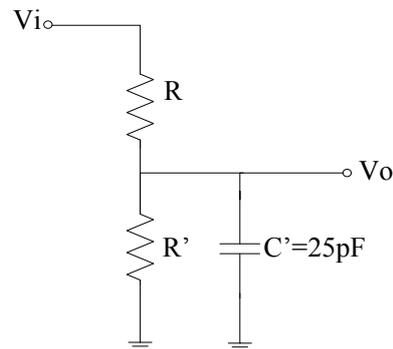


Figura 19 Divisor de voltaje resistivo junto con capacidad equivalente de osciloscopio

Esta capacidad C' , a la salida de la etapa, nos produce una deformación en la señal debido a la formación de un filtro paso bajo RC, cuya constante de tiempo ζ es elevada, debido a la gran atenuación buscada (relación entre R y R'). En este caso, tendremos una gran atenuación en altas frecuencias lo que resultará perjudicial en nuestro estudio.

Otros parásitos, como la inductancia del layout y de los propios componentes no se han considerado en el análisis inicial. La razón es el poder establecer un modelo simple que opere correctamente en el BW de interés. Los valores estimados de los componentes L_s y del layout resonarán a frecuencias altas, que se encuentran muy lejos de nuestro rango de BW.

Como solución al filtro paso bajo RC que se forma en la salida de la etapa, se introduce una compensación, que tendrá como propósito conseguir una respuesta plana del sistema, a lo largo de todo el ancho de banda. En caso contrario, las frecuencias elevadas se verán bien amplificadas (sobrecompensación) o bien atenuadas (subcompensada). Se trata de un circuito ecualizador simple. Dicha compensación consiste en añadir cierta capacidad en paralelo con la resistencia a la entrada, quedando nuestro circuito atenuador como se representa en la Figura 20.

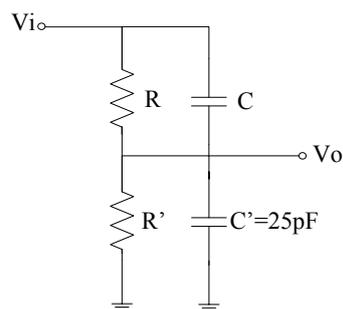


Figura 20 Divisor de voltaje resistivo con compensación de capacidad de osciloscopio

Para conseguir que la compensación sea efectiva, se deberá cumplir la siguiente igualdad:

$$C * R = C' * R'$$

La razón por la que debe cumplirse dicha igualdad se explicará a continuación. Teniendo dos divisores de tensión, por un lado uno resistivo y por otro uno capacitivo:

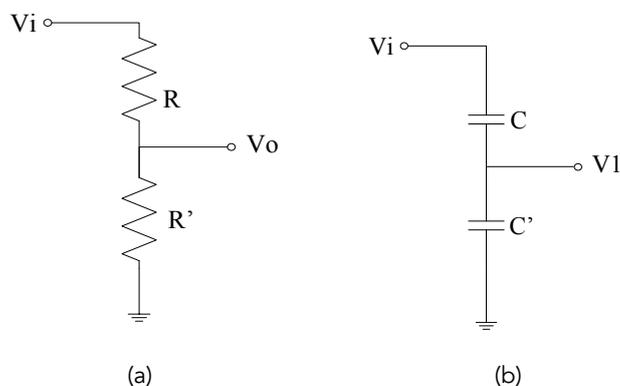


Figura 21 Divisor de tensión resistivo (a) y Divisor de tensión capacitivo (b)

Buscamos que en el primero ocurra:

$$\frac{V0}{Vi} = \frac{R'}{R + R'}$$

Y en el segundo lo siguiente:

$$\frac{V1}{Vi} = \frac{C}{C + C'}$$

Si conectamos ambos en paralelo, vemos que V1 Y V0 son iguales siempre que se cumpla que:

$$C * R = C' * R'$$

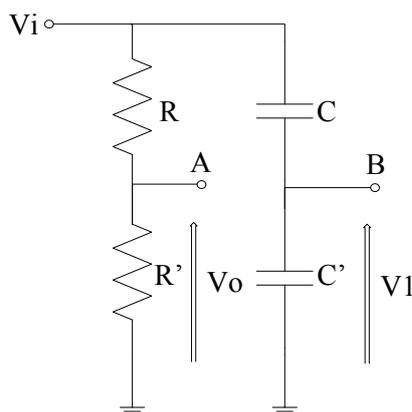


Figura 22 Divisores de tensión en paralelo

Por lo tanto el nodo A y B podrán conectarse, cuando se cumpla dicha CONDICIÓN DE EQUILIBRIO.

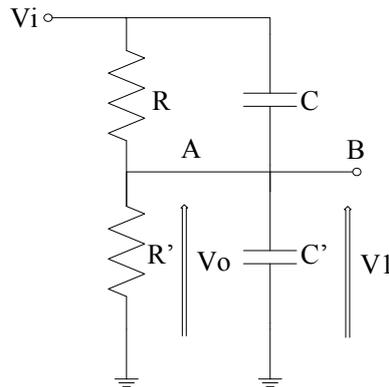


Figura 23 Divisores de tensión en paralelo conectando A y B

A continuación se obtiene la función de transferencia entre Vi y Vo. A partir de ella, aplicando las condiciones de equilibrio se deberá demostrar que la compensación genera un BW alto.

En base al circuito anterior, y trabajando ahora en términos de impedancias, para considerar el efecto de la frecuencia, se tiene que la condición necesaria para que el circuito esté compensado en frecuencia es que la impedancias mostradas en la Figura 24 sean iguales.

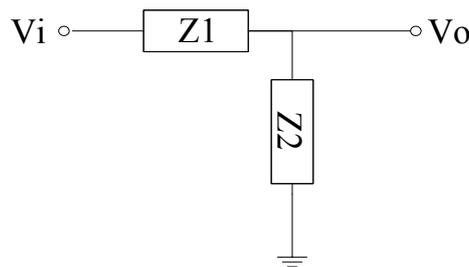


Figura 24 Divisores de tensión en función de impedancias

En este punto, vemos el valor de Z1 Y Z2.

$$Z1 = \frac{R * XC}{R + XC} = \frac{R(\frac{1}{j\omega C})}{R + (\frac{1}{j\omega C})} = \frac{R}{1 + j\omega RC}$$

$$Z2 = \frac{R' * XC'}{R' + XC'} = \frac{R'(\frac{1}{j\omega C'})}{R' + (\frac{1}{j\omega C'})} = \frac{R'}{1 + j\omega R' C'}$$

Se propone utilizar la variable auxiliar $G(w)$ para expresar el factor de atenuación en términos de impedancia:

$$G(w) = \frac{Z2}{Z1 + Z2} = \frac{\frac{R'}{1 + jwR'C'}}{\frac{R}{1 + jwRC} + \frac{R'}{1 + jwR'C'}}$$

En este punto, si aplicamos la CONDICIÓN DE EQUILIBRIO ($RC=R'C'$) , podremos ver como se produce igualdad de denominadores, y por lo tanto la variable $G(w)$, deja de depender de la frecuencia.

$$G(w) = \frac{Z2}{Z1 + Z2} = \frac{\frac{R'}{1 + jwR'C'}}{\frac{R}{1 + jwRC} + \frac{R'}{1 + jwR'C'}} = \frac{\frac{R'}{1 + jwR'C'}}{\frac{R}{1 + jwR'C'} + \frac{R'}{1 + jwR'C'}} = \frac{R'}{R + R'}$$

Queda demostrado así, la independencia de la función de transferencia con la frecuencia de trabajo.

En estas circunstancias, el atenuador es independiente de la frecuencia (BW infinito) y de la forma de onda de la señal y la relación entre entrada-salida es la siguiente:

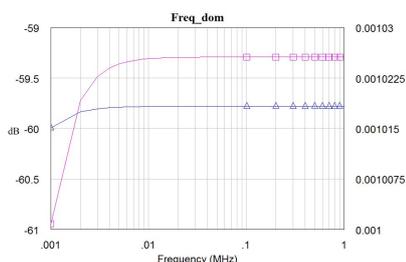
$$\frac{VO}{Vi} = \frac{1}{a}$$

Siendo el valor de a la atenuación producida.

Sin embargo, puede dejar de cumplirse la condición de equilibrio, lo que supondría una respuesta de atenuación no plana en frecuencia.

Cabria contemplar dos casos, donde la línea azul representa la atenuación entre puerto de entrada y salida en dB.

$RC > R'C'$: Se produce atenuación menor en las frecuencias altas (sobrecompensación)



$RC < R'C'$: Las frecuencias altas son más atenuadas de lo previsto. (subcompensación)

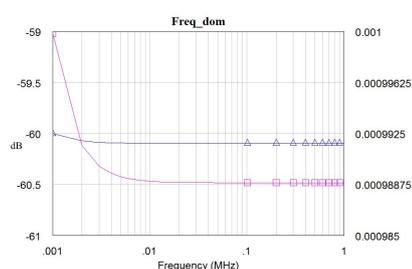


Figura 25 Sobrecompensación y Subcompensación en frecuencia.

Ahora realizaremos el diseño de valores para nuestro acoplador.
Queremos factor 1000 de atenuación, por lo que la relación entre resistencias será:

$$\frac{R'}{R} = \frac{1}{1000}$$

Eligiendo: $R' = 10k\Omega$, $R = 10M\Omega$.

Si aplicamos la condición de equilibrio que necesitamos, siendo $C' = 25\text{ pF}$:

$$C * R = C' * R'$$

$$C = \frac{C' * R'}{R} = \frac{25\text{pF} * 10k\Omega}{10M\Omega} = 0,025\text{ pF}$$

Observamos que el valor de C obtenido es muy pequeño, por lo que deberemos encontrar cierta variación en la topología para que cada uno de los valores que pongamos en nuestro diseño sea coherente y adecuado.

Para conseguir que la capacidad de C, sea mayor, dado que deberán cumplir una relación factor 1000 con C' , necesitamos establecer C' mayor de 25pF. Para ello, añadimos en paralelo a C' cierto condensador que aumentara la capacidad en ese punto (C_t).

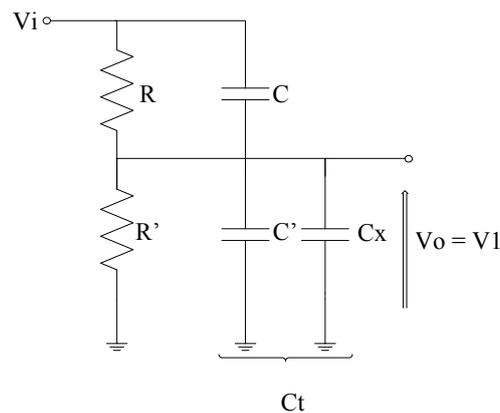


Figura 26 Atenuador con C_x adicional

Establecemos como valor apropiado $C_t = 100\text{ nF}$.
Por lo tanto:

$$C = \frac{C_t * R'}{R} = \frac{100\text{nF} * 10k\Omega}{10M\Omega} = 100\text{pF}$$

Siendo el valor de C todavía bajo y por lo tanto aun no aceptable, establecemos diversas etapas en la primera parte de nuestro atenuador, siendo n el número de etapas posibles.

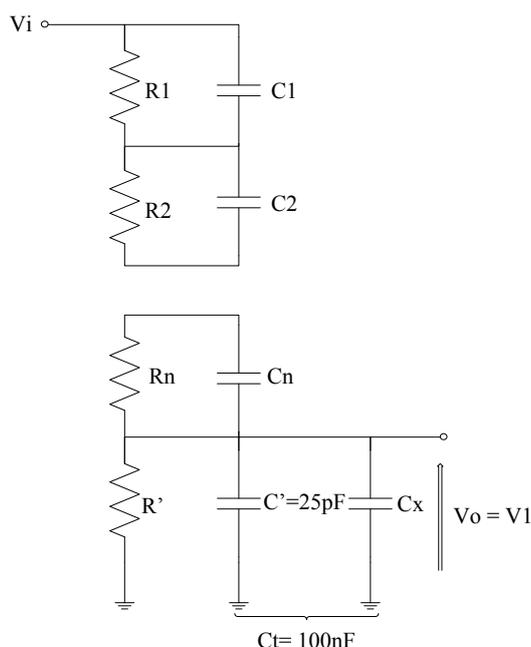


Figura 27 Topología de acoplador n niveles

Calculamos el valor de los componentes, siendo n el número de niveles total de nuestro atenuador:

$$R_n = \frac{R}{n - 1}$$

$$C_n = (n - 1) C$$

$$C' = C_t - C_x = C_t - 25 \text{ pF}$$

Se considera válida y eficiente la topología del acoplador con n niveles en el modelo electrónico que la forma. A continuación se estudia el valor ptimo de niveles a implementar.

3.2.- Simulación

Con la ayuda del software MicroWave Office, se ha verificado la respuesta de varios tipos de topologías con diferentes números de etapas y componentes comerciales (modelos en valores obtenibles comercialmente).

Se introduce en ambos casos una señal cuadrada con amplitud 15000 V en alto y -15 000 V en bajo (señal ideal que llegaría al tejido del paciente).

3.2.1.- Topología 5 niveles.

En primer lugar se calcula el valor de los componentes, siendo n el número de etapas total de nuestro atenuador, en este caso $n=5$.

$$R_n = \frac{R}{n-1} = \frac{10\text{ M}\Omega}{5-1} = 2.5\text{ M}\Omega$$

$$C_n = (n-1)C = (5-1)100\text{ pF} = 400\text{ pF}$$

$$C' = C_t - C_x = C_t - 25\text{ pF} = 100\text{ nF} - 25\text{ pF} \cong 100\text{ nF}$$

Por lo tanto el modelo circuital del atenuador a simular es el mostrado en Figura 28.

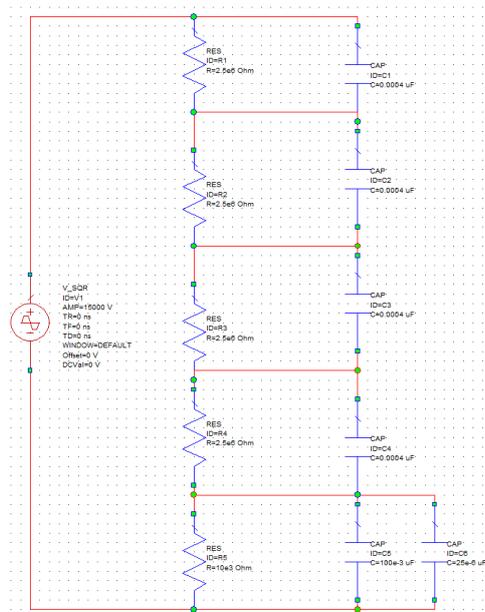


Figura 28 Modelo simulado de topológica acoplador 5 niveles

Comparando la señal de entrada al circuito con la que se obtiene a la salida (Figura 29), permite observar la atenuación correcta en un factor 1000, y la conservación de morfología.

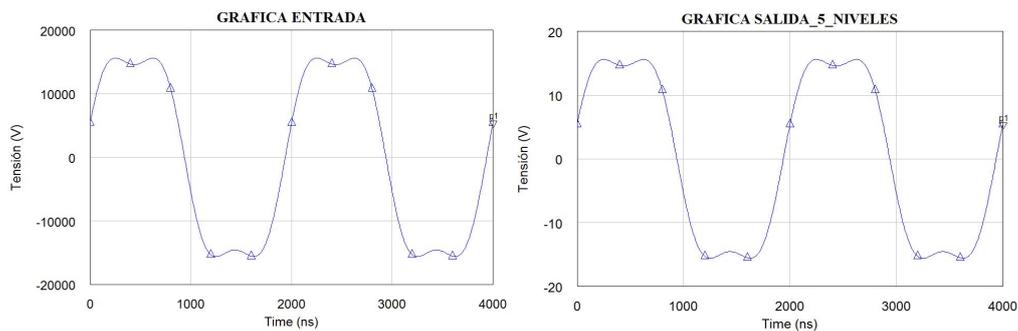


Figura 29 Formas de onda de entrada y salida del acoplador 5 niveles

También simulamos la atenuación en la salida respecto a la entrada en decibelios. Se comprueba la correcta atenuación factor 1000:1, y por tanto atenuación 60dB con respuesta plana en toda la banda de frecuencias simulada (línea rosa).

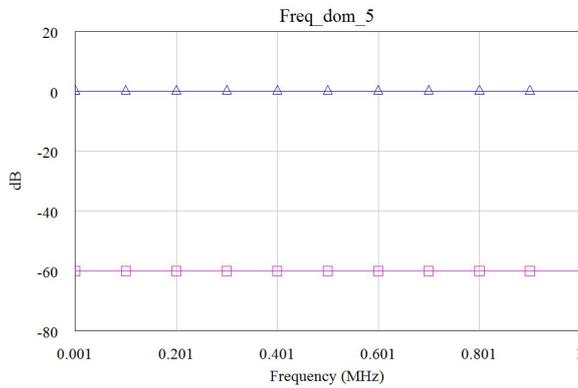


Figura 30 Atenuación ocasionada por el acoplador de 5 niveles.

3.2.2.- Topología 11 niveles

De nuevo, se calcula el valor de los componentes necesarios para implementar en este caso el atenuador de 11 niveles y por tanto donde $n=11$.

$$R_n = \frac{R}{n-1} = \frac{10\text{ M}\Omega}{11-1} = 1\text{ M}\Omega$$

$$C_n = (n-1)C = (11-1)100\text{ pF} = 1\text{ nF}$$

$$C' = C_t - C_x = C_t - 25\text{ pF} = 100\text{ nF} - 25\text{ pF} \cong 100\text{ pF}$$

Con los valores apuntados, el montaje del atenuador quedará de la siguiente forma:

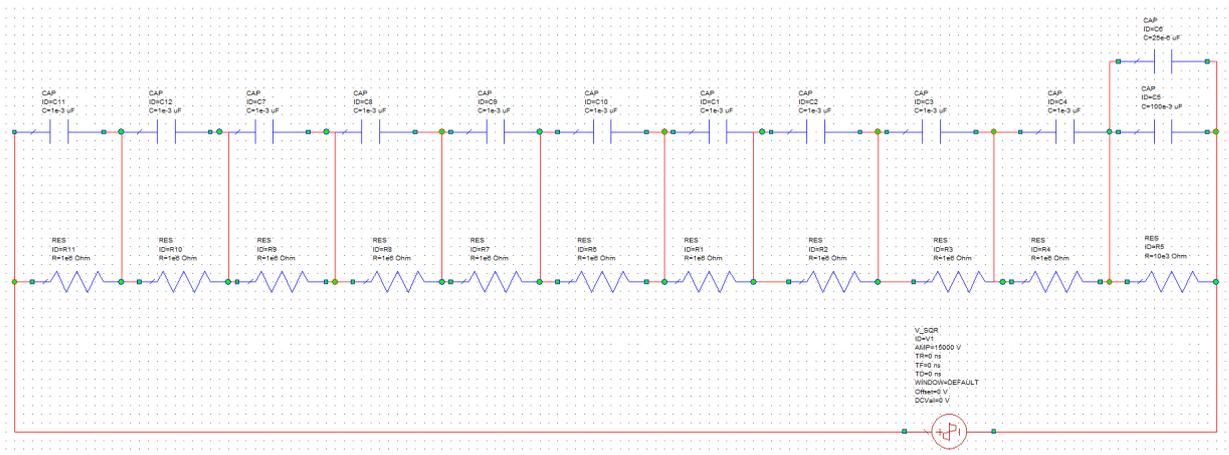


Figura 31 Modelo simulado de topológica acoplador 11 niveles

Para verificar la atenuación que aplica el diseño, se compara la señal en los extremos del circuito (entrada y salida).

Se puede observar el factor 1000 de atenuación en la amplitud de voltaje y correcta captación de la señal, ya que no se produce deformación en forma de onda.

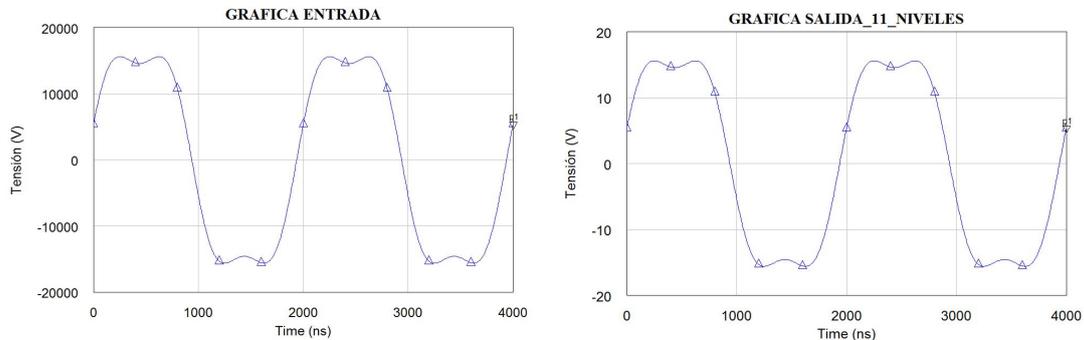


Figura 32 Formas de onda de entrada y salida del acoplador 11 niveles

Del mismo modo, se realiza el estudio de atenuación entre puertos en función de la frecuencia de trabajo. Se observa una respuesta plana a 60dB de atenuación en la Figura 33 (línea rosa), consiguiendo la compensación total, provocada por el cumplimiento de la condición de equilibrio.

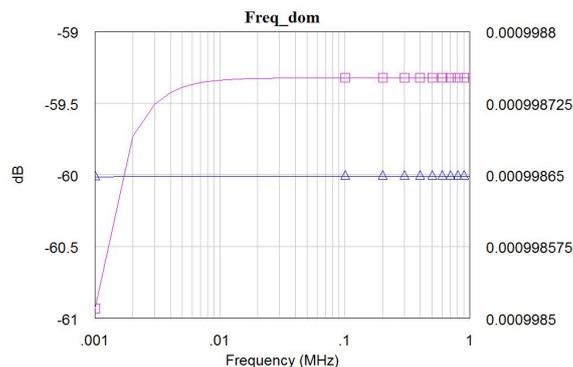


Figura 33 Atenuación ocasionada por el acoplador de 11 niveles.

Una vez estudiadas las simulaciones con los dos tipos de topología, el siguiente paso es elegir una de ellas para el posterior diseño en PCB y puesta en marcha. Conocemos y debemos tener en cuenta la existencia de parásitos en el diseño, para su compensación, será favorable un valor de resistencias menor y por el contrario valores mayores de capacidad. Esta premisa, nos hace decantarnos por el segundo tipo de topología, la de 11 niveles en el modelo circuital.

3.3.- Diseño y montaje sobre PCB

Una vez verificado en simulación el correcto funcionamiento de nuestro diseño, se desarrolla en el entorno Eagle el diseño de placa de circuito impreso (PCB). La impresión en PCB de nuestro circuito permitirá posteriormente el testeo real del sistema.

En el proceso de diseño, se debe tener en cuenta factores que permitan un correcto funcionamiento y por tanto así asegurarnos que el diseño cumple con los requerimientos y limitaciones tecnológicas. Se tendrá en cuenta además que se tratará de componentes de montaje superficial (SMD).

- **Guarda (Clearance):** El espaciado elegido y especificado por los fabricantes, entre los pads y pistas. Manteniendo esta distancia entre pistas se reduce el riesgo de cortocircuitarlas debido a imperfecciones en la fabricación de la PCB o debido a un exceso de estaño en la soldadura. Por otro lado, las pistas (sobre todo las que llevan mucha corriente, como en nuestro caso) se inducen mutuamente. Este efecto, que es no deseable, debe evitarse separando las pistas, en la PCB diseñada se ha dejado distancias en torno a los 7 mm.
- **Creepage:** Distancia entre trozos de cobre a través del sustrato. Se considerará respetar la distancia necesaria para evitar arcos eléctricos, bajo el componente, por contaminación. Esta restricción será determinante en la elección del package de los componentes. Se han escogido packages 2512, que mantendrá creepage de unos 6 mm.
- **Distance:** Se define también la distancia entre los elementos de cobre en el diseño y el borde exterior. En nuestro caso, dejaremos distancias mínimas de 2.5 cm de los componentes y pads al borde, y 1 cm del conector BNC al borde. Evitaremos de este modo arcos eléctricos ente placa y caja, en el sistema electroporador.
- **Ancho de pista (Width):** El espesor mínimo de las pistas de cobre lo establecemos en 0.056 mm. Este tamaño de pista permite reducir el valor de inductancia parasita implícita en el sistema.

En la elección de componentes, se busca obedecer a ciertas especificaciones que los mismos deberán cumplir.

Siendo la elección de topología final, la formada por 11 niveles, deberemos tener en cuenta la tensión nominal que caerá en cada una de esas etapas, y por lo tanto será la que como mínimo necesitará soportar cada condensador.

Suponiendo tensión continua en la entrada, alcanzando los 15000 V, se podría esperar que en cada una de las etapas cayesen unos 1500 V. Por lo cual, en las especificaciones del componente, deberá indicar que soporta una tensión mayor a la esperada.

Lo mismo ocurre en cuanto a las resistencias, la potencia a soportar en cada etapa de las 11 existentes, deberá superar la que encontraremos en la puesta en marcha del dispositivo, que corresponderá con la potencia media del sistema.

Sabiendo que **Potencia Media** :

$$\bar{P} = P_{inst} * duty$$

Es necesario por un lado, conocer la Potencia Instantánea que genera el acoplador (P_{inst}), y por otro tener en cuenta el ciclo de trabajo del pulso que tratamos ($duty$).

Cálculo de Potencia Instantánea:

$$P_{inst} = \frac{V^2}{R} = \frac{15000^2}{10^6} = 225 \text{ W}$$

Cálculo de Duty:

$$Duty = \frac{t_{on}}{T}$$

Se podrán considerar dos casos:

- Single Pulse → Como se ha descrito anteriormente, en este tipo de pulso, cada secuencia de señal tiene duración de 1 segundo, que se corresponde con el periodo T. De este segundo, el pulso ocupa 20 us (periodo de pulso), por lo tanto $t_{on} = 10us$.

$$Duty_{single} = \frac{t_{on}}{T} = \frac{10us}{1s} = \frac{10 * 10^{-6}}{1} = 10^{-5}$$

- Burst Pulse → En esta ocasión, en cada secuencia de señal de 1s (T), se podrán emitir más de 1 pulso. Consideraremos el caso mas desfavorable, en el cual se escoge el máximo de pulsos a aplicar al paciente. Este valor de worst-case será 500 pulso. Considerando como en el anterior, 10 us de trabajo en cada pulso, tendremos $t_{on} = 500 * 10us = 5000us$.

$$Duty_{burst} = \frac{t_{on}}{T} = \frac{5000us}{1s} = \frac{5000 * 10^{-6}}{1} = 5 * 10^{-3}$$

Por lo tanto, valorando los dos tipos de pulsos, encontraremos dos resultados de Potencia Media que afectarán a todo el circuito.

$$\bar{P}_{single} = P_{inst} * duty = 225 * 10^{-5} = 2.25 \text{ mW}$$

$$\bar{P}_{burst} = P_{inst} * duty = 225 * 5 * 10^{-3} = 1.125 \text{ W}$$

Conocido este dato, se deberá tener atención en la elección de las resistencias. Analizando el peor caso, si nos encontramos en pulso Burst, cada resistencia deberá disipar como mínimo:

$$\bar{P}_{resistencia} = \frac{1.125}{10} = 0.1125 \text{ W}$$

En el ANEXO 2, encontramos la elección y descripción de los componentes usados en la PCB final.

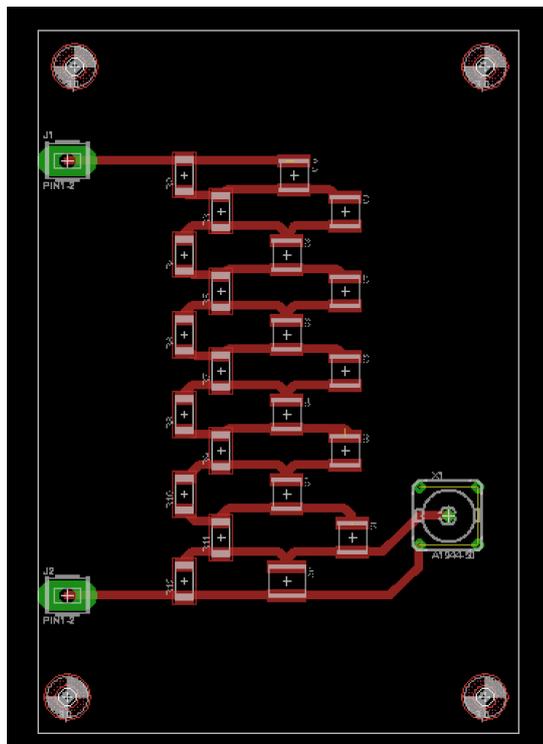


Figura 34 Diseño PCB en el entorno EAGLE

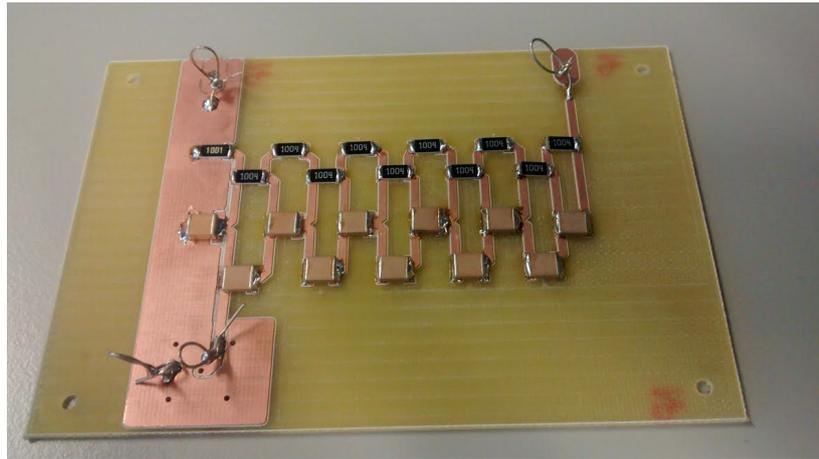


Figura 35 Placa PCB acoplador 11 niveles

3.4.- Testeo y verificación

3.4.1.- Análisis frecuencial en pequeña señal

La herramienta clave en este proceso es el Portable Analog Circuit Design Kit (ANALOG DISCOVERY) que DIGILENT ofrece.



Figura 36 ANALOG DISCOVERY, DIGILENT

Con el uso del software WaveForms, proporcionado por DIGILENT de nuevo, nos permite realizar el análisis espectral buscado.

En cuanto a los elementos hardware de los que hacemos uso, además del propio ANALOG DISCOVERY, requerimos uno de los complementos que proporciona el mismo fabricante, el Analog Discovery BNC Adapter Board.



Figura 37 BNC Adapter Board

Esta última herramienta, permite el uso de cables de prueba terminados en BNC estándar y sondas en el ANALOG DISCOVERY.

Las conexiones entre los diferentes elementos quedarían de la siguiente forma.

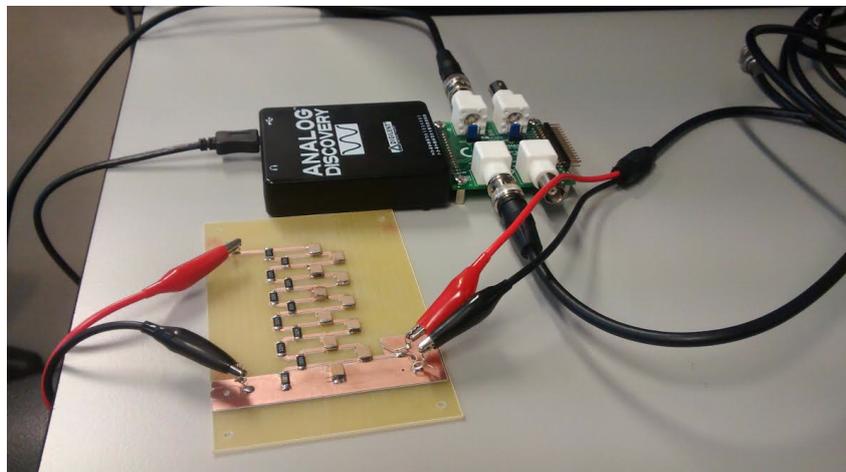


Figura 38 Conexiones de simulación

Por el CHANNEL 1 del osciloscopio digital veremos la salida del atenuador. En cuanto a la entrada del mismo, será creada digitalmente por el software utilizado, pero introducida de forma analógica para la verificación real de la atenuación producida. Dicha señal de entrada se captará en el W1 del Adapter Board.

Dentro del entorno WaveForms, la herramienta específica Network Analyzer nos permitirá comprobar la atenuación a la salida en función de la frecuencia. El máximo voltaje que la aplicación admite sacar en DC será 5V, y el barrido de frecuencias que se puede analizar va desde 10KHz a 10 MHz.

Integración de control, medida y ensayos de un electroporador para tratamientos quirúrgicos

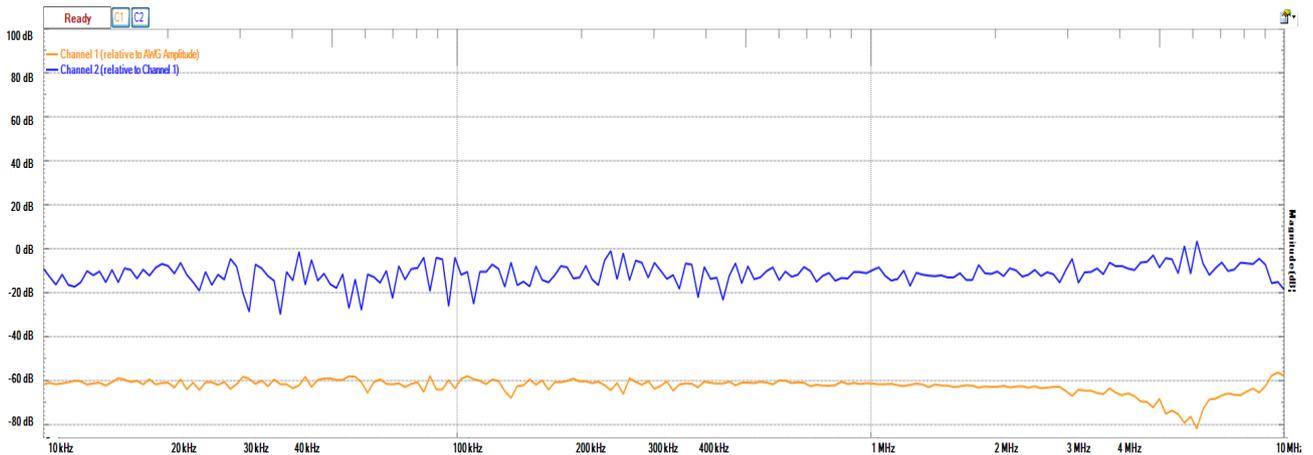


Figura 39 Atenuación producida por el acoplador diseñado

La compensación en frecuencia que aplicamos en el diseño de nuestro atenuador se ve reflejada en el análisis mostrado en la Figura 39 (línea naranja), dado que prácticamente en todas las frecuencias presentadas vemos como permanece la atenuación de tensión en factor 1000:1 (-60 dB).

Sin embargo, si que podemos considerar ciertos puntos del análisis donde dicha atenuación no se mantiene del todo constate.

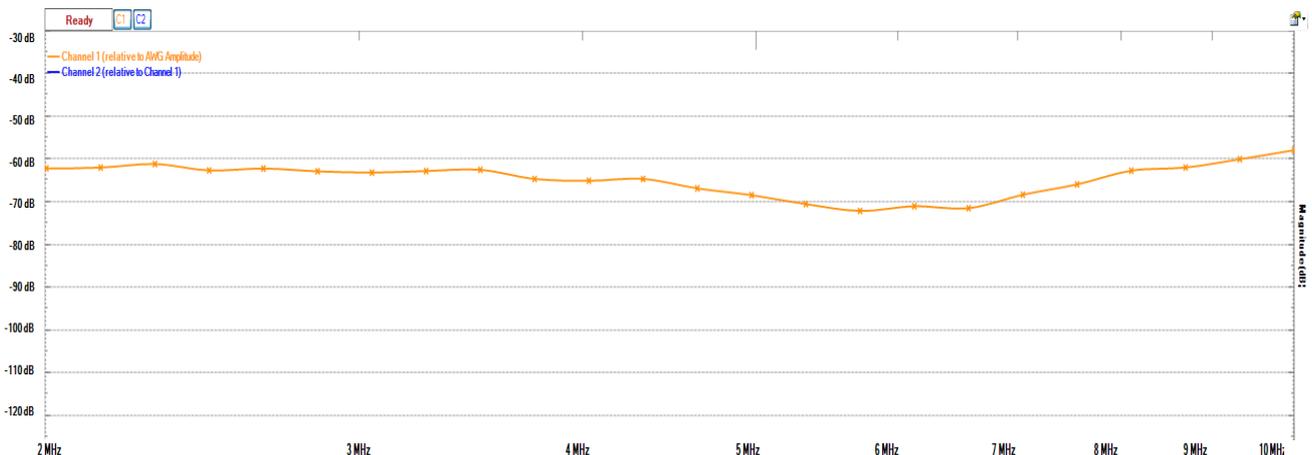


Figura 40 Zoom Atenuación producida por el acoplador diseñado

Haciendo zoom en el análisis, como se ve en la Figura 40, podemos observar como en las más altas frecuencias aparece una descompensación en frecuencia y la respuesta deja de ser plana en ese BW. La atenuación es superior a la estimada, por lo tanto implica que el acoplador se encuentra subcompensado, como consecuencia de $RC < R'C'$.

Otro factor a tener en cuenta será el ruido que se introduce en los cables del osciloscopio. Tras la atenuación de 5 V a 5 mV, la señal, será muy sensible a contaminación de ruido.

3.4.2.- Análisis en gran señal

En este apartado se estudiará el funcionamiento del acoplador diseñado con entradas de pulsos cuadrados de amplitud 6kV. Para ello se han utilizado las primeras pruebas del conjunto completo, que no incluye la carga biológica (entorno de quirófano).

En la Figura 41, podemos visualizar la comparación de las formas de onda a la entrada y a la salida del acoplador. La segunda, ha sido amplificada para conseguir establecer una superposición y conclusiones concluyentes.

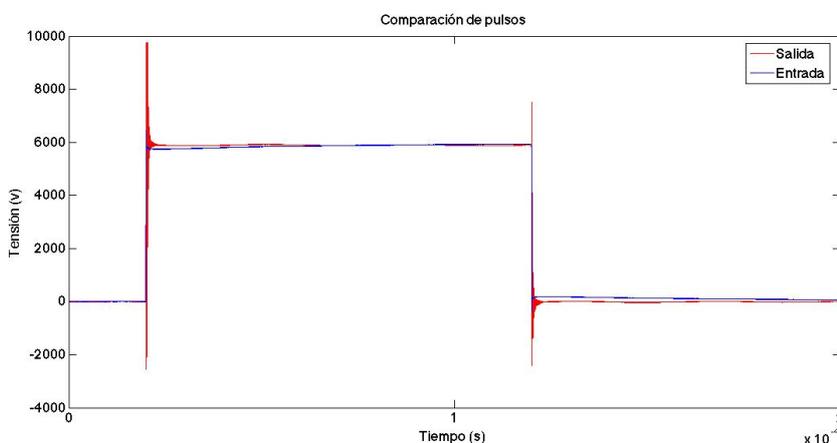


Figura 41 Comparación de pulsos de entrada y salida del acoplador

La Figura 41, en la que se superponen dichas señales, muestra que no existe sobre oscilación o degradación de la forma de pulso achacable al BW.

Sin embargo, si que se observa una sobreoscilación de alta frecuencia (en torno a los 2.5 MHz) al comienzo del techo de pulso, como se reproduce en la Figura 42, que se estima proveniente de la resonancia parásita del circuito implementado.

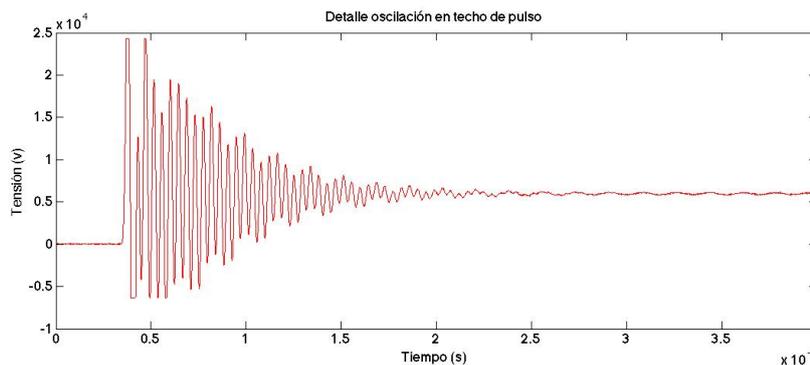


Figura 42 Zoom Oscilación de techo de pulso a la salida del acoplador

3.4.3.- Análisis en DC

En esta ocasión se inyecta un valor fijo de tensión a la entrada del atenuador que nos es proporcionado por un generador de señal. Como herramienta clave para esta tarea se utiliza un multímetro de mesa de alta precisión.

Aumentando progresivamente, en este caso, el nivel de voltaje, se obtienen los siguientes resultados:

Vin	15,279 V	20,4 V	30,401 V	40,303V	50,233V
Vout	15,236 mV	20,325mv	30,316 mV	40,153mV	50,172mV

Tabla 1 Datos análisis DC

Como consecuencia de lo observado, se puede demostrar su correcto funcionamiento en DC.

3.4.3.- Caracterización completa del sistema atenuador

Una vez tenemos el montaje soldado en la PCB diseñada, podemos verificar si en cada punto de la placa tenemos los valores de capacidad y resistencia deseados. A continuación se presenta de nuevo el esquema planteado del acoplador de 11 niveles, con dichos valores medidos con el uso de nuevo de un multímetro de mesa.

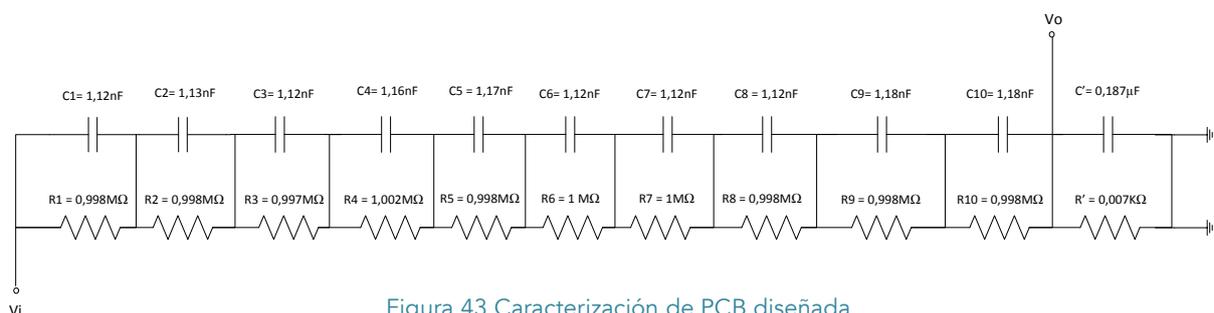


Figura 43 Caracterización de PCB diseñada

Para un estudio completo en función de los valores reales que podrían tomar los componentes dada su tolerancia, se decide elaborar un análisis estadístico . Este permitirá conocer el efecto en la atenuación esperada que produce la alteración del valor nominal del componente.

Se plantean dos tipos de análisis, *Worst-Case* y *Monte Carlo*. Se considera un 1% y un 5% de tolerancia en las resistencias y condensadores respectivamente. Y con la herramienta PSpice OrCAD se simulan ambos análisis, y se procede a la importación a Matlab para su correcta representación y visualización.

Análisis Worst-Case: Calcula la respuesta del acoplador culminando en un único "run", donde los valores de los componentes del modelo se han puesto en el peor caso posible, valores extremos en función de su tolerancia. Primer caso, con valores máximos que podrían alcanzar los componentes y en el segundo, con los valores mínimos que la tolerancia de un componente permitiría tener.

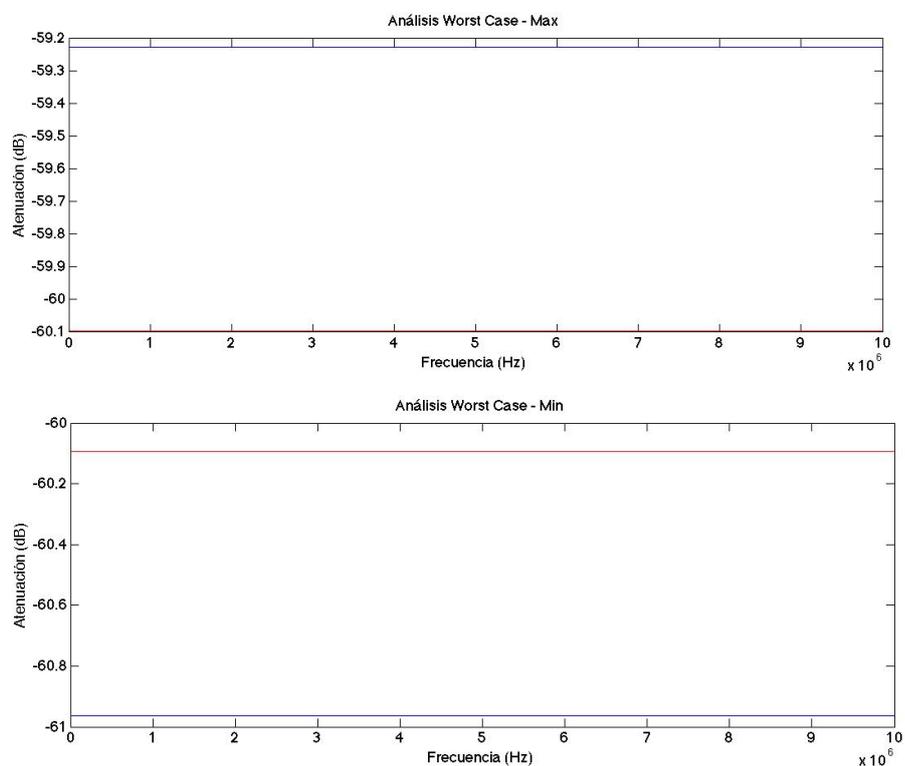
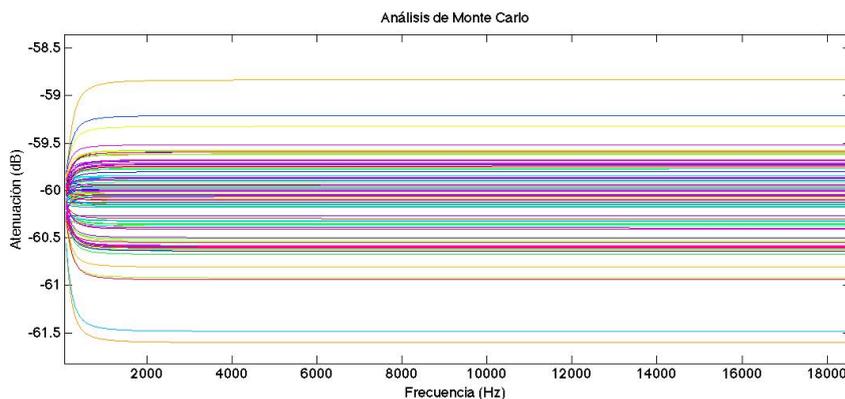


Figura 44 Análisis Worst-Case para las tolerancias de los componentes

Análisis Montecarlo: Donde calcula las variaciones que experimentara la respuesta del diseño ante cambios aleatorios en los valores de los componentes en función de su tolerancia.



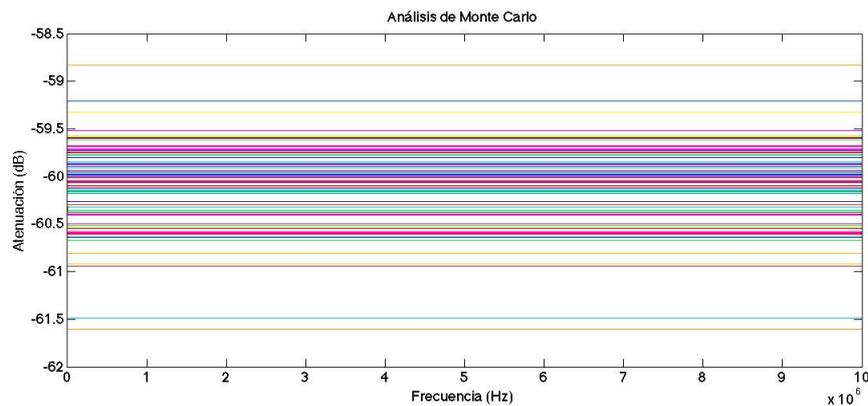


Figura 45 Análisis de MonteCarlo para las tolerancias de los componentes.

Dado el realismo y aproximación a la realidad, el Análisis Montecarlo nos proporciona un estudio más completo. Y además podremos concluir que la variación de los valores de componentes respecto a su tolerancia, únicamente afecta en la atenuación en un factor escalar. Este factor, como se puede apreciar en las representaciones, como máximo será de 1.5 dB (de exceso o defecto). Sin embargo, la respuesta de atenuación en función de la frecuencia sigue siendo plana en toda la banda exceptuando frecuencias bajas. Ya que nuestro sistema actuará e en torno a los 50kHz, esta parte transitoria no nos afecta a nuestro funcionamiento.

3.4.4.- Mejoras y modificaciones

Dado que el valor de la capacidad encontrada en la última etapa del atenuador difiere con la esperada, deberemos buscar que modificación en dicha etapa final, lo podría solucionar. Teniendo 187 nF y esperando tener 100nF, se produciría la descompensación en frecuencia (subcompensación). Esta capacidad extra se añade por la cercanía del plano de masa, necesario para conectar el instrumento de medida con la pista del último nivel de atenuación (salida del acoplador).

Esto hace que no tengamos ancho de banda infinito en cuanto a la obtención de los buscados -60dB, y la necesidad que encontrar la forma de volver a tener,

$$C * R = C' * R'$$

Si $C' = 187\text{nF}$, implica que $C * R < C' * R'$, por lo que deberemos disminuir el valor a la derecha de la desigualdad.

Dado que en los valores testeados existentes en la PCB creada, se tiene, $C * R = 1,198 * 10^{-3}$, si $R' = 9,997\text{k}\Omega$, se calcula el valor que se desearía encontrar en C' .

$$C' = \frac{C * R}{R'} = 1.198 * 10^{-7}$$

Si colocamos un condensador auxiliar, C_{aux} , en serie con C' , conseguiremos disminuir la capacidad que medimos en dicho punto.

$$C_{t2} = \frac{1}{\frac{1}{187} + \frac{1}{330}} = 119,4 \text{ nF}$$

Lo que permitirá que, $C' * R' = 119,4 * 10^{-3} \cong C * R$.

Por lo que, se consigue una compensación real de la frecuencia, si añadimos en el montaje un condensador de capacidad 330 nF en serie con C' en la etapa de salida.

Vemos la representación de la PCB en la Figura 46, donde se ha añadido el condensador auxiliar que compensará de nuevo el sistema.

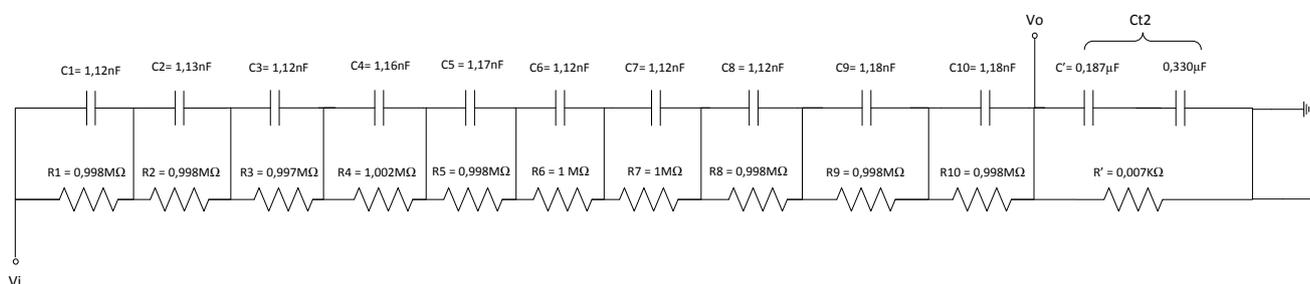


Figura 46 Acoplador real compensado

4 Creación de Software de control

4.1.- Objetivo

La herramienta de software a implementar se enmarca en el contexto del proyecto en curso sobre la electroporación irreversible de células cancerígenas.

Se pretende desarrollar un proyecto software de control básico que permita, al personal cualificado, seleccionar el campo eléctrico que aplicará en tejido afectado. El alcance de esta aplicación pretende simular en pequeña escala el sistema final que tendrá el médico en la sala de operación o de tratamientos.

Se usa para la implementación soportes dedicados al aprendizaje, que nos proporcionan salidas analógicas de máximo 5 V. Dicho voltaje dista al necesario en el sistema profesional, que estaría en torno a los cientos de voltios aplicados en el primario del transformador, pero el desarrollo de este software abarca por primera vez la parte de control del usuario final. Como ya se ha especificado, se necesitan aplicar pulsos cuadrados, por lo que la implementación deberá mostrar y crear los mismos.

Para su desarrollo se siguen los pasos del modelo lineal-secuencial para la ingeniería de software. Se establecen los siguientes bloques.

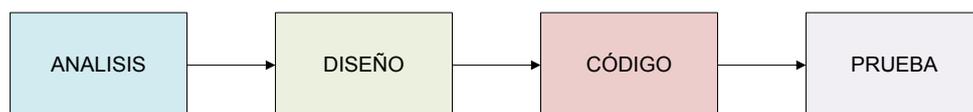


Figura 47 Bloques de software

4.2.- Ingeniería de Software: Análisis

En este apartado se llevará a cabo el proceso de reunión de requisitos que deberá tener la aplicación. Se tiene muy presente hacia quien va dirigida la aplicación y sus funcionalidades.

Como requisitos principales, la posibilidad de elegir los parámetros que conformarán las formas de onda y la creación de un medio con el cual el usuario disparará la señal cuando él mismo lo elija.

Los parámetros fundamentales que constituye la forma de onda son, el número de pulsos que se lanzarán el forma de ráfaga cada segundo y su ciclo de trabajo.

Además, se abarcará la forma de comunicación remota entre el dispositivo que utiliza el medico en el tratamiento y el sistema generador de ondas.

4.3.- Ingeniería de Software: Diseño y Código

Se adjunta en el ANEXO 3 el código completo del programa desarrollado[5].

Se ha elegido como microprocesador para la implementación del software, el que proporciona el microcontrolador Arduino Due, el SAM2X8E. Dicho microprocesador tiene múltiples periféricos, pero en este caso hacemos uso de uno de ellos, los *timers*.

El sistema timer es muy completo y con el uso de funciones permite generar una onda cuadrada PWM y a la vez ejecutar síncronamente y con prioridad una trozo de código de programa (código de interrupción). Es precisamente lo que deseamos que ejecute nuestro sistema, pulsos de tensión cuadrados.

El microcontrolador consta de 9 canales de timer, de los cuales erigiremos el 7. Este canal es parte del timer principal TC2, en concreto se tratará del subcanal 1 del TC2.

Cada timer es un contador de 32 bits que se incrementan a una frecuencia de reloj submúltiplo del reloj principal que trabaja a 42MHz.

Elegimos preescalar la frecuencia principal con un factor 2. Este es el factor más pequeño que permite la arquitectura de distribución de reloj del SAM32X8E. Y queremos que la onda cuadrada que crea el sistema sea de $f_s=8000\text{Hz}$. Dicha frecuencia es menor de la que realmente queremos alcanzar en un sistema perfeccionado, pero usando 8000Hz facilita la tarea de visualizado.

Para ajustar la frecuencia de la onda cuadrada que deseamos reproducir, será necesario configurar el registro RC. La obtención del valor RC, nos la proporciona la siguiente ecuación.

$$RC = \frac{42 \text{ Mhz}}{n} * \frac{1}{f_s}$$

Siendo $n=2$, el factor preescala y $f_s=8000\text{Hz}$, frecuencia de onda cuadrada.

Una vez ajustada la frecuencia de la onda que se crea, es necesario usar y configurar registros que permitirán su generación, estos son RA y RB. Cada registro, está asociado a un pin de salida. Dado que utilizamos el timer 7, RA está asociado al pin 3 (TIOA3) y RB al pin 10 (TIOB10). Los valores que toman estos pines, será 1 cuando el timer sea menor que el registro y tomará valor 0 cuando timer sea mayor, como vemos en la Figura 48.

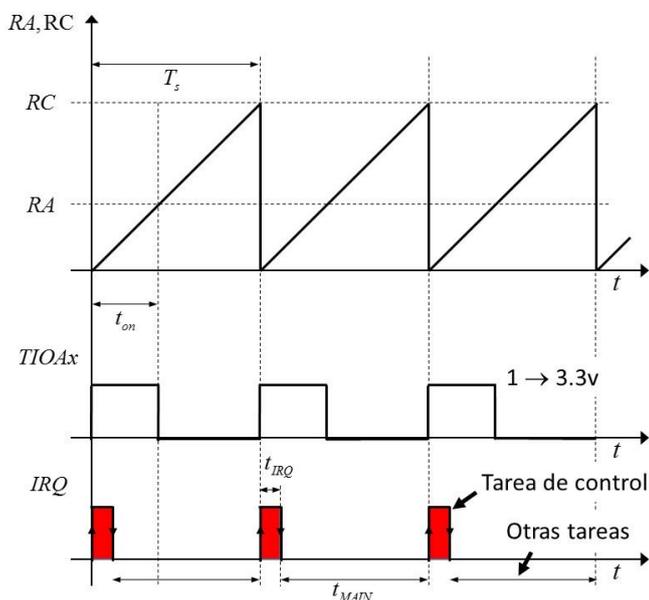


Figura 48 Ejecución de la interrupción y salida de onda cuadrada con el timer

Lo señalado anteriormente implica que modificando los valores de RA, podremos controlar el tiempo de activación de la señal de control (t_{on}) manteniendo el periodo fijo y solamente dependiente de RC.

Como podemos ver de forma concreta y detallada en el ANEXO 3, se hace uso de la interrupción elegida para ejecutar nuestro código de control cada vez que se inicie la onda cuadrada.

Para conseguir plasmar las características deseadas en la onda cuadrada, deberemos modificar el valor de los registros RA y RB, el cual cargará el t_{on} seleccionado. Nuestro objetivo es conseguir que el tiempo en alto sea la mitad del periodo total de la onda.

El valor de los registros se obtiene de igual modo que RC:

$$RA = \frac{42 \text{ Mhz}}{n} * \frac{1}{fs} = \frac{42 \text{ Mhz}}{2} * (2 * tiempo_{on}) = 42\text{Mhz} * tiempo_{on}$$

$$RB = (42\text{Mhz} * tiempo_{on}) + 1\text{us}$$

Las formas de onda obtenidas en los registros RA y RB serán complementarias entre sí en la entrada al driver del sistema. Para compensar el tiempo de cambio en el puente inversor, se añade 1ms de diferencia al registro RB que permitirá controlar el tiempo muerto (retraso entre la conmutación de una rama y la adyacente en el inversor).

El tratamiento de electroporación busca aplicar pulsos de onda cuadra con periodos de duty 0.5. Aún así, en la implementación de software se le da al usuario posibilidad de seleccionar el ciclo de servicio del canal.

Del mismo modo, las especificaciones en el tratamiento indican que se debe poder controlar el número de pulsos aplicados al tejido tumoral, ya que excesivos, serían inadecuados para el proceso. Para ello, el software desarrollado permite especificar el número de pulsos que se emitirán (máximo 500).

Cabe destacar, que el software desarrollado cubrirá ciertas funciones de seguridad básicas. Se trata de la implementación de medidas que fueron expuestas en el bloque Desacoplador-descargador desarrollador por el proyecto de Jorge Izano [6]. En dicho sistema se incluyeron relés adicionales que añaden medidas de seguridad del sistema en caso de fallo. En concreto, se pretende implementar funcionalidades para el sistema Soft-Start y relés de seguridad (“mains” y “crowbar”).

Relé soft-start

El sistema *Soft-start* consiste en un relé y una resistencia de precarga. Su misión es la de controlar la intensidad de carga del condensador (*inrush current*) de forma que no se produzca un pico de intensidad excesivo. Hay distintas formas de funcionamiento posible. En general se carga el condensador a través de la resistencia de precarga durante cierto tiempo (por ejemplo, hasta el 60% de carga máximo) y entonces se cierra el relé cortocircuitando la resistencia. De esta forma se evitan las pérdidas de conducción en la resistencia de precarga.

El valor de la resistencia se calcula con una especificación de la intensidad máxima y con la tensión máxima posible entre sus bornes. En nuestro caso se estiman 2 segundos de carga de condensador.

Relé de crowbar

Como sistema último de seguridad se incluye un relé entre alimentación y tierra junto con un fusible antes del relé. De esta forma si el relé de *mains* fallara se cerraría el relé *crowbar*. Esto crearía un cortocircuito y provocaría un aumento muy considerable en la intensidad. El fusible entonces actuaría y cortaría la alimentación. Se crea, por lo tanto, una forma definitiva de corte de la alimentación en caso de que el *mains*, de menor orden de seguridad, falle y no deje el circuito en abierto. Se estimarán 2 segundos antes de hacer la comprobación del relé *mains*.

4.3.1.- Firmware

Se definirán en este bloque el conjunto de instrucciones y procedimientos del programa desarrollado. Este conjunto, fija la lógica primaria que ejercerá el control sobre el microcontrolador Arduino Due.

1. Inicialización de variables
 - 1.1. Variables I/O analógicas y digitales.
 - 1.2. Variables asociadas al timer.
 - 1.3. Variables intermedias de formas de onda
 - 1.4. Variables destinadas al command parser.
2. Declaración de timer-interrupción (*startTimer*)
3. Inicialización valor de registros
4. Activación y habilitación de interrupción
5. Main (setup)
 - 5.1. Configuración de pines
 - 5.2. Inicialización de puerto serie
 - 5.3. Reserva de 200 bytes de string de entrada
 - 5.4. Inicialización comandos disponibles en el command parser.
 - 5.5. Habilitación e inicialización de periféricos.
 - 5.6. Llamada e inicialización de timer (*startTimer*)
6. Main (loop). Reconocimiento de comando recibido y ejecución de función correspondiente
7. Declaración de funciones requeridas en parsing-command (*DISP()*,*ON()*,*OFF()*,*PW()*,*PN()*,*CROWBAR()*).
8. Rutina de interrupción (*TC7_Handler()*)
9. Declaración función Serial (*serialEvent*)

Además, a continuación en la Figura 49, se representan los diagramas de flujo del programa implementado.

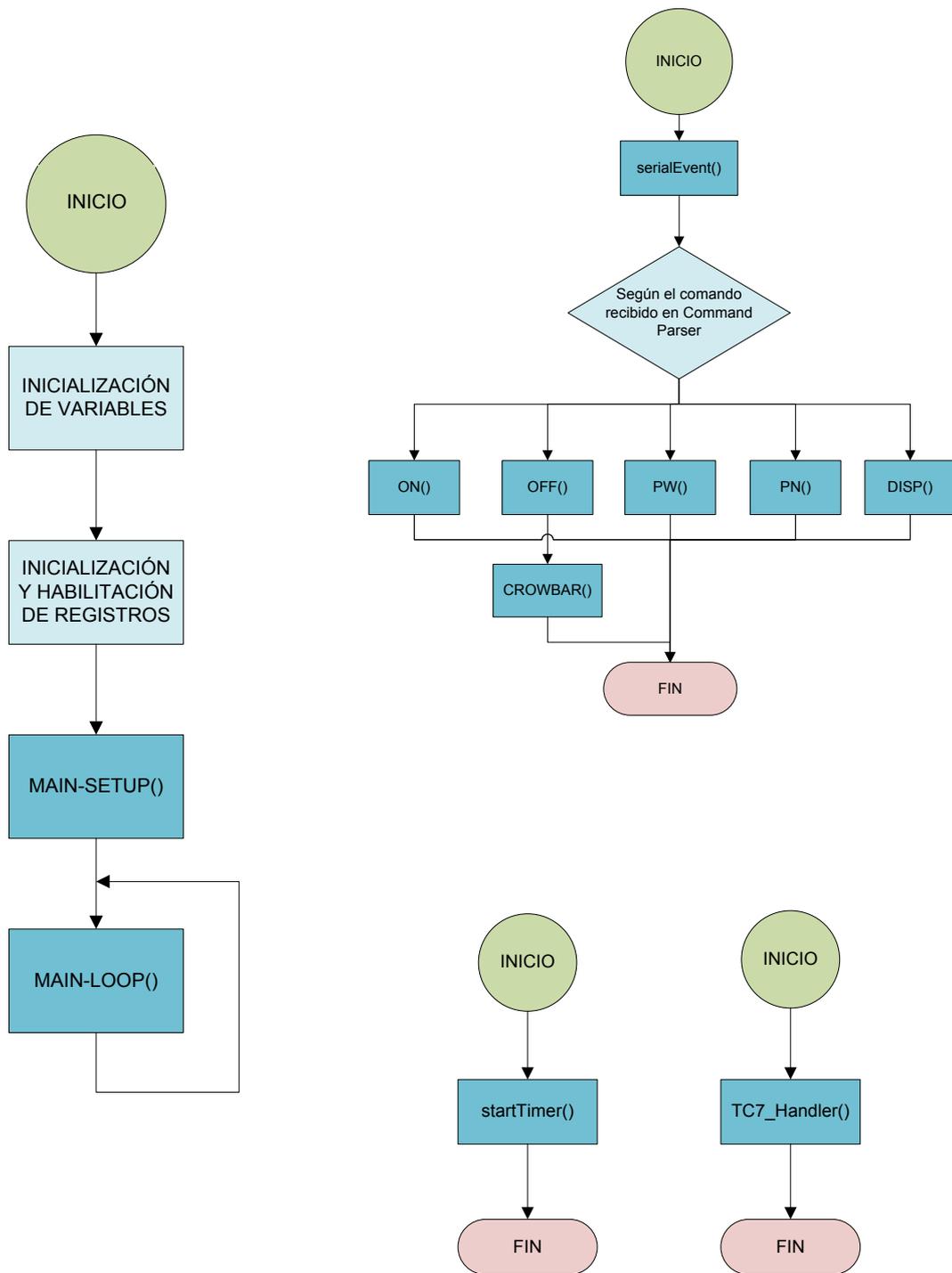


Figura 49 Diagramas de flujo del programa implementado en Arduino Due

Se detalla a continuación en el Tabla 2, el desglose y descripción de variables principales utilizadas.

NOMBRE	TIPO	DESCIPCIÓN
entrada_crowbar	int	Permite leer la entrada analogía procedente de la señal aplicada al paciente. Es utilizada en la implementación del relé crowbar.
analogPin	int	Selecciona el pin relacionado con la entrada entrada crowbar.
frecuencia	uint_32t	Frecuencia seleccionada para la onda de salida cuadrada.
canal	uint_32t	Representa el subcanal usado dentro del timer principal correspondiente, que a su vez se selecciona en la variable Tc*chTC.
Tc*chTC		Interpreta el timer principal usado.
rc	uint_32t	Registro que permite ajustar la frecuencia de onda cuadrada.
num_pulsos, num_pulsos_count	int	Simboliza el número de pulsos que conformará la señal de salida.

tiempo_on	int	Permite el manejo de ciclos de trabajo de la forma de onda.
inputString	string	Almacena los datos de entrada por el puerto serie.
stringComplete	boolean	Permanecerá en falso hasta que el string no haya terminado.
cmd_disp_a, cmd_disp_t, cmd_on, cmd_off, ack,conectado	string	Strings posible que reconocerá el programa.
encendido	boolean	Permite conocer que el usuario pone en activo el sistema. Solo cuando este en true, podrá manejar las demás funcionalidades.
habilitado	boolean	Concede la posibilidad de añadir seguridad al sistema. Solo se podrá acceder a las funciones del sistema si está en true.

Tabla 2 Detalle de variables principales

4.3.2.- Protocolo de comunicación

El propósito final es la comunicación de nuestro microcontrolador con un dispositivo que en último término usaría el medico cualificado. Pero antes de afrontar la comunicación se deberá construir el protocolo que lo permita.

La comunicación del usuario con el software que se ha implementado se basa en la implementación de un *Command Parser*. Este, se sustenta en el uso de puerto serie (inicializado en el programa), para la recepción de comandos tipo string. Dicha comunicación se podrá establecer a través del Monitor Serie que proporciona el entorno Arduino, o a través de comunicación Bluetooth haciendo uso de los pines RX1, TX1 que se comunican directamente con el puerto serie inicializado.

Los comandos enviados, serán reconocidos por el software, y en función del identificado por el programa, será ejecutada la función asociada al mismo.

Se muestra un diagrama de flujo a continuación que muestra el protocolo de actuación.

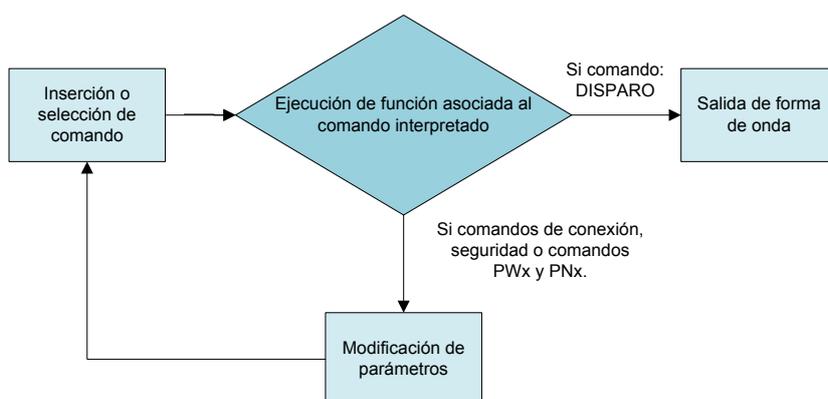


Figura 50 Diagrama flujo protocolo comunicación

Comandos de conexión y seguridad.

“ON”- La aplicación se encuentra inicializada y puesta a punto para su uso. La selección del comando también ilumina un led verde para indicarlo. Inmediatamente tras la activación de este comando, se interpreta el encendido del sistema y se pone en marcha el protocolo de seguridad denominado Soft-Start. Este, permite esperar un tiempo de 2 segundos, y tras ese retraso, el programa estará totalmente habilitado para su uso.

“OFF”- Implica la desactivación del software. Cabe destacar que en este proceso se llevará a cabo también protecciones en el sistema. Dichas funciones de seguridad se comenzarán a ejecutar en el preciso instante en el que se inicie la función. Tras dos segundos desde que se selecciona la opción de apagado, por un lado, se deshabilitará el programa completo resolviendo la seguridad Soft-start, y por otro, se comprobará cual es el nivel de tensión que se esta siendo aplicando en ese momento. Si pasados dos segundos tras la desconexión, el nivel de señal supera un cierto umbral (que seleccionaremos), deberán cortocircuitarse las salidas por las que se conforma la onda (pines 3 y 10).

Comandos de datos y aplicación.

“PWx”- El usuario podrá seleccionar el tiempo en alto de la onda cuadrada. Para ello tras el comando PW seguida un valor x, el cual deberá oscilar entre 10 a 100. Este valor x hace referencia al número de microsegundos que deseamos tener el pulso en alto.

“PNx”- De igual forma, el especialista podrá escoger el número de pulsos que aplicar al paciente. Deberá aparecer el comando PN seguido de un valor x, que podrá oscilar entre 1 y 500 pulsos.

“DISPARO”- Una vez elegidos los valores que construirán la onda siguiendo las especificaciones del tratamiento concreto, se podrá iniciar la emisión de la onda. Solo cuando el sistema este encendido y habilitado por la función de seguridad Soft-start, se permitirá cargar los valores de RA y RB y al saltar la interrupción la onda cuadrada aparecerá con los valores por los que se ha optado.

4.3.3.- Comunicación Bluetooth

Como se ha señalado anteriormente, se pretende abordar la comunicación necesaria entre el dispositivo controlador que especifica y emite los pulsos necesarios para el tratamiento y el usuario del prototipo.

Tras la implementación de un software que permite la emisión de la señal que se pretende aplicar, se plantea el control de ese software vía bluetooth.

A través del puerto serie, conseguiremos que el dispositivo Arduino Due reciba los comandos implementados en el programa. En la etapa de pruebas, hacíamos uso del monitor serial que proporciona el entorno Arduino, pero lo que se busca ahora es el manejo de este puerto serie (para comunicarnos con nuestro microcontrolador) vía bluetooth.

Hacemos uso de los pines digitales 0 (RX0) y 1 (TX1) que permiten la comunicación con el puerto serie del Arduino.

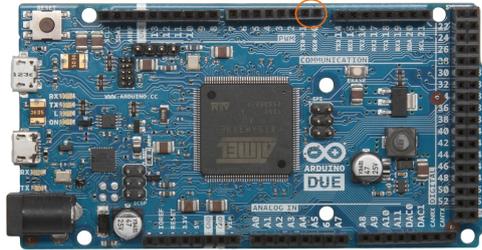


Figura 51 Microcontrolador Arduino Due

Se ha elegido el módulo bluetooth HC-06 que solo puede actuar como esclavo, por lo tanto será el dispositivo móvil o tablet quien se conecte al módulo.

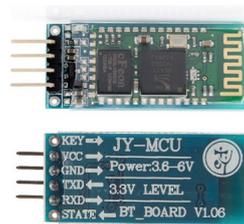


Figura 52 Módulo Bluetooth HC-06 para Arduino

Los pines que encontramos son los siguientes:

VCC: Alimentación del módulo entre 3,6 V y 6V. Lo conectamos a la alimentación de 5V que proporciona en uno de sus pines Arduino Due.

GND: La masa del módulo, que del mismo modo irá conectada a uno de los pines GND que proporciona el microcontrolador.

TXD: Transmisión de datos. Este pin del módulo es conectada con el pin 0 (RX0) de Arduino Due.

RXD: Recepción de datos a un voltaje de 3,3V. Dado que los pines de Arduino Due tienen salida a 3,3 V podremos conectar directamente este pin con el pin 1 (TX1) del procesador.

Los pines KEY y STATE no serán usados en nuestra aplicación.

Una vez tenemos los pines conectados a nuestro Arduino, el módulo bluetooth será capaz de controlar el puerto serie del microcontrolador. Por tanto los datos

que se envíen por el pin TXD del módulo se recibirán, y si coinciden con los comandos implementados, el programa ejecutará su funcionalidad y se emitirán las señales esperadas.

El montaje en placa blanca tendrá la siguiente disposición.

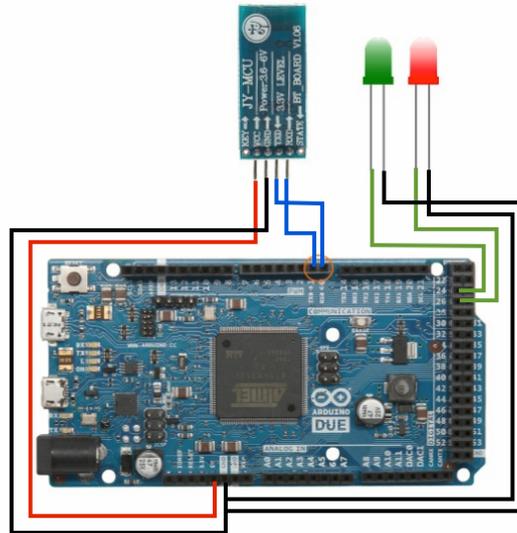


Figura 53 Montaje módulo Bluetooth-Arduino Due

A continuación se aborda la comunicación del bluetooth con otro dispositivo móvil.

4.3.4.- App Android

Para el envío de datos a través de un dispositivo móvil al módulo bluetooth se pueden usar diferentes técnicas. En esta ocasión se ha elegido la producción de una aplicación para dispositivos Android. Esta consigue controlar y comunicarse con el programa que corre en Arduino a través de comunicación exclusivamente bluetooth.

AppInventor2 es un entorno de desarrollo de aplicaciones para dispositivos Android, se trata de una herramienta de programación basada en descripción de modelos por bloques con generación automática de código. La elección de su uso se trata en el ANEXO 4.

App Inventor Designer, donde se construye el interfaz de usuario. En nuestra aplicación deben aparecer una serie de componentes básicos que nos permitan manejar los comandos que controlan el software que corre en Arduino. Se describen los componentes detalladamente en ANEXO 5.



Figura 54 Interfaz de usuario en la App Android

CONECTAR A DISPOSITIVO: Una vez abierta la aplicación, el usuario será capaz de visualizar los dispositivos vinculados y disponibles en ese momento por comunicación bluetooth. En nuestro caso, el usuario deberá seleccionar el usuario que coincida con el módulo conectado al Arduino. Para una sencilla visualización de si se ha conseguido establecer esa conexión entre los dos dispositivos, si la conexión ha sido exitosa, aparecerá una notificación comunicando al usuario de ello.

Botón ON: Una vez vinculado, el usuario podrá encender el sistema. Tras ser pulsado, el usuario deberá esperar a que aparezca y más tarde desaparezca una notificación indicando que el sistema esta habilitado. Así mismo, se ilumina el led verde con la misma indicación.

Botón OFF: Si el usuario desea apagar el sistema y que no sea posible el envío de pulsos al paciente, deberá pulsar este botón. Tras ser pulsado volverá a aparecer una notificación donde informa que el sistema ha sido deshabilitado totalmente.

Despliegue NÚMERO DE PULSOS: El usuario podrá desplegar una pestaña donde le aparecen todas las posibilidades de cantidad de pulsos a enviar. Una vez seleccionada la opción que desea, debe pulsar el Botón APLICAR, para confirmar su decisión.

Despliegue TIEMPO EN ON: El usuario podrá desplegar una pestaña donde podrá seleccionar el tiempo que desea tener en alto la onda cuadrada. Podrá seleccionar un valor que vaya desde 10 a 100. Al pulsar el Botón APLICAR, se aplicará su selección.

Botón DISPARAR: Solo cuando el usuario se haya conectado al dispositivo bluetooth, haya puesto en ON el sistema habilitandolo , y haya seleccionado tanto numero de pulsos como el tiempo que desea que el pulso este en alto, deberá pulsar DISPARAR. En este momento el programa implementado en Arduino Due creará y emitirá las ondas cuadradas.

App Inventor Blocks, donde se define el comportamiento de los componentes que componen la interfaz de la aplicación. Se trata de programación por bloques. Se hace uso de integrados de control, lógica, procedimientos y texto, para conseguir la conducta adecuada de la aplicación. Se especifica un descripción detallada de bloques, variables e instrucciones en el ANEXO 5.

CONECTAR_DISPOSITIVO: Al pulsar el botón, deberemos conseguir dos utilidades básicas:

- Mostrar la lista de usuarios *bluetooth* disponibles:

```
when CONECTAR_DISPOSITIVO .BeforePicking
do
  if BluetoothClient1 . Available
  then
    set CONECTAR_DISPOSITIVO . Elements to BluetoothClient1 . AddressesAndNames
```

- Establecer conexión con el dispositivo seleccionado e indicar en una notificación que la conexión ha tenido éxito.

```
when CONECTAR_DISPOSITIVO .AfterPicking
do
  evaluate but ignore result call BluetoothClient1 . Connect
  address CONECTAR_DISPOSITIVO . Selection
  if BluetoothClient1 . Available
  then
    call NOTIFICACIÓN_CONECTADO . ShowAlert
    notice uppercase " EXISTE CONEXIÓN CON EL DISPOSITIVO BLUETOOTH "
```

ON: Cuando el usuario pulsa el botón ON, siempre que exista conexión bluetooth con otro dispositivo, se enviará el comando ON que se recibirá en el puerto serie de Arduino. Además para implementar la función de seguridad Soft-start, se codifica con un reloj un retraso de 2 segundos, hasta que el sistema es habilitado completamente para su uso.

```
when ON .Click
do
  set VISTA_MONITOR . Text to " ON "
  if BluetoothClient1 . Available
  then
    call BluetoothClient1 . SendText
    text " ONn "
  set RELOJ_HAB . TimerEnabled to true
  call HABILITADO . ShowAlert
  notice uppercase " HABILITADO "
  set RELOJ_HAB . TimerEnabled to false
```

OFF: Cuando el usuario pulsa OFF, como ocurre cuando se pulsa ON, solo enviara el comando OFF a Arduino si se encuentra en conexión habilitada con el módulo bluetooth. De nuevo, se implementa el relé Soft-Start, con la ayuda del retraso de 2 segundos que asegurará el correcto funcionamiento del sistema.

```

when OFF.Click
do
  set VISTA_MONITOR.Text to " OFF "
  if BluetoothClient1.Available
  then
    call BluetoothClient1.SendText
      text " OFF\n "
    set RELOJ_DESHAB.TimerEnabled to true
    call DESHABILITADO.ShowAlert
      notice uppercase " DESHABILITADO "
    set RELOJ_DESHAB.TimerEnabled to false
  
```

APLICAR_PULSOS: Una vez el usuario seleccione del desplegable el número de pulsos que desea aplicar, deberá pulsar APLICAR y podremos ver aparecer el valor en la pequeña pantalla que emula el monitor serial de Arduino. Además, la implementación enviará los comandos para la comunicación con el puerto serie de Arduino Due, en este caso PNx, siendo x número de pulsos a emitir.

```

when APLICAR_PULSOS.Click
do
  set VISTA_MONITOR.Text to NUMERO_PULSOS.Selection
  if BluetoothClient1.Available
  then
    call BluetoothClient1.SendText
      text " PN "
    call BluetoothClient1.SendText
      text NUMERO_PULSOS.Selection
    call BluetoothClient1.SendText
      text "\n "
  
```

APLICAR_TIEMPO: Del mismo modo que en el caso anterior, tras seleccionar el ciclo de trabajo del pulso, el usuario deberá confirmar su elección pulsando APLICAR. Al ser pulsado este botón podremos ver en el emulador de monitor serial la elección. Además se enviarán al microcontrolador los comandos adecuados, en esta ocasión PWx, siendo x el valor en microsegundos del pulso en alto.

```

when APLICAR_TIEMPO.Click
do
  set VISTA_MONITOR.Text to TIEMPO_ON.Selection
  if BluetoothClient1.Available
  then
    call BluetoothClient1.SendText
      text " PW "
    call BluetoothClient1.SendText
      text TIEMPO_ON.Selection
    call BluetoothClient1.SendText
      text "\n "
  
```

DISPARAR: Tras realizar los pasos anteriores, el usuario únicamente tendrá que pulsar DISPARAR para que su selección se haga efectiva y Arduino cree la onda cuadrada implementada.

Si el bluetooth ha sido desconectado, y no existe vinculación entre dispositivo y módulo bluetooth el led que indicaba conexión se apagará. Además el usuario podrá ver un mensaje de alerta que se lo indicará. En este punto, deberá volver a establecer conexión.

```
when DISPARAR .Click
do
  if BluetoothClient1 . Available
  then
    call BluetoothClient1 .SendText
    text "DISPARO\n"
    set VISTA_MONITOR . Text to "DISPARO "
  else
    call BluetoothClient1 .SendText
    text "NOCONECTADO\n"
    set VISTA_MONITOR . Text to "ALERTA: NO CONECTADO "
```

LIMPIAR: Botón auxiliar para eliminar el texto que aparezca en el emulador de monitor serial en ese momento.

```
when LIMPIAR .Click
do
  set VISTA_MONITOR . Text to ""
```

Para el uso del CLOCK ,que permitirá, habilitar y deshabilitar 2 segundos después de encender y apagar la aplicación. Debemos mantenerlo deshabilitado cuando se inicie la pantalla, y no se habilitarán hasta no pulsar ON u OFF.

```
when Screen1 .Initialize
do
  set RELOJ_HAB . TimerEnabled to false
  set RELOJ_DESHAB . TimerEnabled to false
```

4.4. Ingeniería de Software: Pruebas y Mantenimiento

Un vez tenemos el sistema de comunicación ANDROID-ARDUINO resuelto se realizan pruebas para comprobar su correcto funcionamiento.

Nos es útil de nuevo la herramienta de DIGILENT, ANALOG DISCOVERY. A través de ella, usando la herramienta OSCILLOSCOPE que proporciona el entorno WaveForms, conseguiremos ver la forma de onda que elabora Arduino Due.

Teniendo disponibles dos canales de entrada en Analog Discovery BNC Adapter Board (CHANNEL 1, CHANNEL 2), usaremos ambos para representar las formas de onda que se obtienen en los pines 3 y 10 ,como se ha especificado anteriormente.

El primer paso a seguir es abrir la aplicación y conectar nuestro dispositivo Android al módulo bluetooth HC-06. Debemos ver que aparece una notificación indicando que la conexión se ha establecido Figura 54.

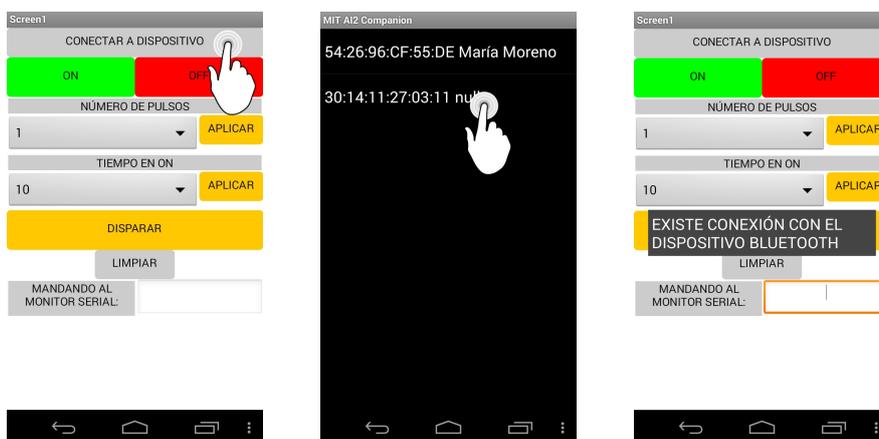


Figura 55

Tras ver aparecer el mensaje anterior podremos poner en ON el sistema con tan solo pulsar la tecla que corresponde. Figura 55.



Figura 56

Integración de control, medida y ensayos de un electroporador para tratamientos quirúrgicos

Se iluminará el led verde de montaje y tras dos segundos, aparecerá el mensaje HABILITADO, que permitirá al usuario continuar usando la herramienta.

Lo siguiente que deberá hacer la persona que use la aplicación será seleccionar en los dos desplegables tanto el número de pulsos, como el tiempo en conducción que desea que conforme el pulso. Figura 57.

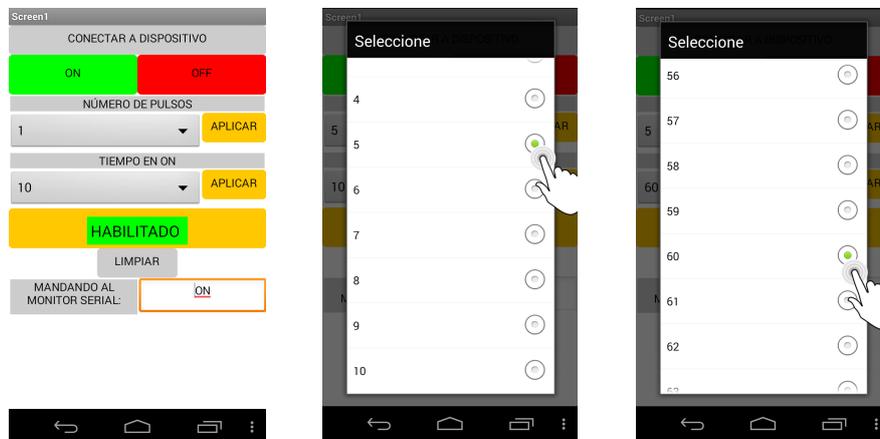


Figura 57

Solo se hará efectiva la selección cuando el usuario pulse aplicar ambos valores, será cuando el Arduino reciba dicha información. Figura 58.



Figura 58

Por último, si el bluetooth no ha perdido su conexión con el dispositivo, podremos enviar los datos elegidos al microcontrolador y este se encargará de crear la forma de onda adecuada. Figura 59.



Figura 59

Visualizando los dos canales que proporciona el osciloscopio digital, podemos capturar la onda de salida tras pulsar DISPARAR en la interfaz de la aplicación. La salida que creará los dos pines correspondientes a los registros (pin 3 y pin 10), puede verse en la Figura 60. Como vemos, la onda naranja corresponde con el registro de salida RA y la azul con el RB. Se comprueba que ambos son complementarios, y con la topología adecuada que precisa la entrada del Driver del sistema electroporador.

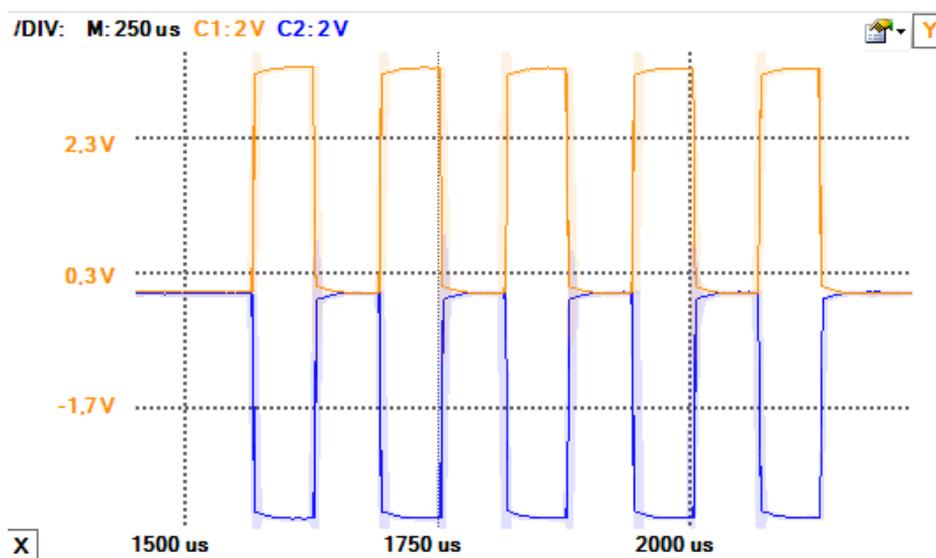


Figura 60 Visualización en el osciloscopio de la forma de onda creada por Arduino Due

Dado que el t_{on} seleccionado ha sido 60ms, cabía esperar que el duty fuese prácticamente 0.5 ya que cada periodo de pulso son 125ms (8000Hz de frecuencia).

Integración de control, medida y ensayos de un electroporador para tratamientos quirúrgicos

Por último, si el usuario desea, inhabilitar y apagar el equipo, deberá pulsar el botón OFF (se enciende led rojo), y del mismo modo que en el proceso de encendido, se ejecutará la función de seguridad Soft-start a los dos segundos y aparecerá una notificación informando de la deshabilitación de la aplicación (DESHABILITADO).
Figura 61.



Figura 61

5 Conclusiones

Se ha realizado un estudio de los pulsos de una electroporación irreversible. Son los que se obtienen de la aplicación de pulsos cuadrados de alta tensión en el tejido del órgano enfermo. De este estudio se selecciona el ancho de banda adecuado para la captación de un valor apropiado de energía (90% de la energía total de la señal).

Se ha planteado y diseñado una nueva topología de acoplador, que consigue la captación de la señal tras su aplicación en el paciente. El diseño, permite una atenuación de amplitud de factor 1000, logrando una distorsión muy baja y respuesta plana en la mayor parte de la banda frecuencial, lo cual indica que el sistema funciona de igual forma independientemente de la frecuencia de trabajo. Para su acreditación, se han elaborado ensayos del diseño implementado en PCB, en pequeña señal, gran señal y análisis DC. Además se ha llevado a cabo la caracterización completa del diseño PCB, elaborando también un análisis de MonteCarlo respecto a la tolerancia de sus componentes. Dichos estudios nos permiten indicar, que en posteriores investigaciones, se podría establecer un nuevo modelo de acoplador considerando los parásitos inductivos, y que mejoraría las respuestas del sistema. También se plantea un posible rediseño reduciendo la influencia de la capacidad parásita del *layout* en la zona cercana con el plano de masa de salida.

La parte final del TFG ha consistido en el desarrollo de un software de control sobre un microcontrolador Arduino Due. El programa permite al usuario crear y modificar parámetros de las formas de onda que serán enviadas al driver del sistema electroporador para su posterior uso en el tratamiento de EPI. El manejo de este software se lleva a cabo por comunicación *bluetooth* implementada y a través del manejo de una aplicación Android desarrollada en el presente trabajo. Dicha aplicación se ha implementado usando el entorno App Inventor 2.

Una vez realizadas las pruebas previstas del software, se ha comprobado que la aplicación Android es capaz de configurar los parámetros correctamente y enviárselos al microcontrolador. Cuando Arduino Due recibe los parámetros, crea las formas de onda de manera unívoca y se visualizan correctamente en el osciloscopio digital de DIGILENT. En un futuro, se propone desarrollar una aplicación Android más sofisticada y que abarque nuevos métodos de seguridad.

ANEXO 1 : Código Matlab para el estudio del pulso de electroporación.

Código estudio pulso BURST- CON ELECTROPORACIÓN

```
%% Mariña Moreno Sau
% TFG GRADO TELECOMUNICACIONES 2015

%Cargamos los datos de la forma de onda

datos=load('PulseBurstDatos.mat');
dat=struct2cell(datos);
data=cell2mat(dat);

%Separamos los datos

vector_tiempos=data(1:1,:);
vector_datos=data(2:2,:);
tension_pico=max(vector_datos);
N=length(vector_datos);

fo=50000;
to=1/fo;
fs=100e6;
ts=1/fs;
t= ts:ts:4*to;

f=fftshift(fft(vector_datos)*2/N );
a=2*abs(f);
l=linspace(-1/2,1/2,N)*fs;
figure
plot(l,a);

e_total=ts*sum(a.^2)/2;
v1=(length(a)-1)/2;

for i=1:v1+1

    intervalo=a(v1:v1+i);
    ener_intervalo=ts*sum(intervalo.^2);
    porcentaje_energia=(ener_intervalo/e_total)*100;
    matriz_porcentajes(i,1)=(i)*fs/N; %frecuencias bw
    matriz_porcentajes(i,2)=porcentaje_energia;

end
```

```

%REPRESENTACION ENERGIA/BW

energias=matriz_porcentajes(:,2);
anchos_banda=matriz_porcentajes(:,1);

plot(anchos_banda,energias);
title('BURST PULSE')
xlabel('ANCHO DE BANDA(HZ)');
ylabel('% ENERGIA');

bar(anchos_banda,energias);
title('BURST PULSE')
xlabel('ANCHO DE BANDA(HZ)');
ylabel('% ENERGIA');

```

Código estudio pulso SINGLE- CON ELECTROPORACIÓN

```

%% María Moreno Sau
% TFG GRADO TELECOMUNICACIONES 2015

%Cargamos los datos de la forma de onda

datos=load('PulseSingleDatos.mat');
dat=struct2cell(datos);
data=cell2mat(dat);

%Separamos los datos

vector_tiempos=data(1:1,:);
figure
vector_dat=data(2:2,:);
vector_dat=vector_dat';
plot(vector_dat);
tension_pico=max(vector_dat);

%Hacemos que aparezca un periodo unicamente cada 1 segundo.
%para ello hacemos zero-padding hasta un numero de muestras que
corresponde
%con 1 seg

vector_datos = padarray(vector_dat(1:81),1012419,zeros,'post');
figure
plot(vector_datos);

N=length(vector_datos);

fo=50000;
to=1/fo;
fs=100e4;
ts=1/fs;

f=fftshift(fft(vector_datos)*2/N );

```

```
a=2*abs(f);
l=linspace(-1/2,1/2,N)*fs;
figure
plot(l,a);

e_total=ts*sum(a.^2)/2;
v1=(length(a)-1)/2;

for i=1:v1+1

    intervalo=a(v1:v1+i);
    ener_intervalo=ts*sum(intervalo.^2);
    porcentaje_energia=(ener_intervalo/e_total)*100;
    matriz_porcentajes(i,1)=(i)*fs/N; %frecuencias bw
    matriz_porcentajes(i,2)=porcentaje_energia;

end

%CREAMOS UNA MATRIZ CON MENOS MUESTRAS PARA PODER ESTUDIARLA MEJOR

matriz=zeros(1200,2);
j=1;

for i=1:1200

    matriz(i,1)=matriz_porcentajes(j,1);
    matriz(i,2)=matriz_porcentajes(j,2);
    j=j+400;
end

%REPRESENTACION ENERGIA/BW

energias=matriz(:,2);
anchos_banda=matriz(:,1);
plot(anchos_banda,energias);
title('SINGLE PULSE')
xlabel('ANCHO DE BANDA(HZ)');
ylabel('% ENERGIA');
bar(anchos_banda,energias);
title('SINGLE PULSE')
xlabel('ANCHO DE BANDA(HZ)');
ylabel('% ENERGIA');
```

Código estudio pulso BURST- SIN ELECTROPORACIÓN

```
%% María Moreno Sau
% TFG GRADO TELECOMUNICACIONES 2015

%Cargamos los datos de la forma de onda
```

```

datos=load('PulseBurstDatosSinEP.mat');
dat=struct2cell(datos);
data=cell2mat(dat);

%Separamos los datos

vector_tiempos=data(1:1,:);
vector_datos=data(2:2,:);
tension_pico=max(vector_datos);
N=length(vector_datos);

fo=50000;
to=1/fo;
fs=100e6;
ts=1/fs;
t= ts:ts:4*to;

f=fftshift(fft(vector_datos)*2/N );
a=2*abs(f);
l=linspace(-1/2,1/2,N)*fs;
figure
plot(l,a);

e_total=ts*sum(a.^2)/2;
v1=(length(a)-1)/2;

for i=1:v1+1

    intervalo=a(v1:v1+i);
    ener_intervalo=ts*sum(intervalo.^2);
    porcentaje_energia=(ener_intervalo/e_total)*100;
    matriz_porcentajes(i,1)=(i)*fs/N; %frecuencias bw
    matriz_porcentajes(i,2)=porcentaje_energia;

end

%REPRESENTACION ENERGIA/BW

energias=matriz_porcentajes(:,2);
anchos_banda=matriz_porcentajes(:,1);
figure
plot(anchos_banda,energias);
title('BURST PULSE')
xlabel('ANCHO DE BANDA(HZ)');
ylabel('% ENERGIA');
figure
bar(anchos_banda,energias);
title('BURST PULSE')
xlabel('ANCHO DE BANDA(HZ)');
ylabel('% ENERGIA');

```

Código estudio pulso SINGLE- SIN ELECTROPORACIÓN

% María Moreno Sau

```
% TFG GRADO TELECOMUNICACIONES 2015

%Cargamos los datos de la forma de onda

datos=load('PulseSingleDatosSinEP.mat');
dat=struct2cell(datos);
data=cell2mat(dat);

%Separamos los datos

vector_tiempos=data(1:1,:);
figure
vector_dat=data(2:2,:);
vector_dat=vector_dat';
plot(vector_dat);

%Hacemos que aparezca un periodo unicamente cada 1 segundo.
%para ello hacemos zero-padding hasta un numero de muestras que
corresponde
%con 1 seg

vector_datos = padarray(vector_dat(1:81),1012419,zeros,'post');
figure
plot(vector_datos);
tension_pico=max(vector_datos);

N=length(vector_datos);

fo=50000;
to=1/fo;
fs=100e4;
ts=1/fs;

f=fftshift(fft(vector_datos)*2/N );
a=2*abs(f);
l=linspace(-1/2,1/2,N)*fs;
figure
plot(l,a);

e_total=ts*sum(a.^2)/2;
v1=(length(a)-1)/2;

for i=1:v1+1

    intervalo=a(v1:v1+i);
    ener_intervalo=ts*sum(intervalo.^2);
    porcentaje_energia=(ener_intervalo/e_total)*100;
    matriz_porcentajes(i,1)=(i)*fs/N; %frecuencias bw
    matriz_porcentajes(i,2)=porcentaje_energia;

end

%CREAMOS UNA MATRIZ CON MENOS MUESTRAS PARA PODER ESTUDIARLA MEJOR

matriz=zeros(1200,2);
```

```
j=1;

for i=1:1200

    matriz(i,1)=matriz_porcentajes(j,1);
    matriz(i,2)=matriz_porcentajes(j,2);
    j=j+400;
end

%REPRESENTACION ENERGIA/BW

energias=matriz(:,2);
anchos_banda=matriz(:,1);
plot(anchos_banda,energias);
title('SINGLE PULSE')
xlabel('ANCHO DE BANDA(HZ)');
ylabel('% ENERGIA');
bar(anchos_banda,energias);
title('SINGLE PULSE')
xlabel('ANCHO DE BANDA(HZ)');
ylabel('% ENERGIA');
```

ANEXO 2 : Selección de componentes para PCB del acoplador.

TIPO	VALOR	FABRICANTE	DESCRIPCIÓN	UNIDADES
Resistencia	1M Ω	Bourns (CRM2512)	Película metálica. Potencia de pico 2000 W. Tolerancia 1%.	10
Resistencia	10 K Ω	Bourns (CRM2512)	Película metálica. Potencia de pico 2000 W. Tolerancia 1%.	1
Condensador	1 nF	Kemet (2220)	Tensión nominal 2kV Tolerancia 5%.	10
Condensador	1000nF	Kemet (2220)	Tensión nominal 2kV. Tolerancia 5%.	1
Conector BNC	Hembra	RS (Z13-060-02- 000212000)	Aislado en ángulo recto. 75 Ω	1

Tabla 3 Selección de componentes para la PCB del acoplador

ANEXO 3 : Código Arduino para software de control.

```
//SOFTWARE DE CONTROL BÁSICO
//Electroporador de células tumorales - María Moreno Sau 2014-2015

//Inicialización de variables I/O analógicas y digitales
int entrada_crowbar=0; //ANALOGICA A6
int analogPin = A6;

//Inicialización variables timer
static uint32_t frecuencia = 8000; // de nuestro pulso 50000
static uint32_t canal = 1; // 3 y 10
static Tc *chTC = TC2;
static uint32_t rc = VARIANT_MCK/2/frecuencia; //preescaler

//Inicialización variables intermedias
int num_pulsos = 1; // numero de pulsos seguidos de disparo
int num_pulsos_count = 0; // contador de pulsos
int32_t tiempo_on = 10; // tiempo on en us

//Inicialización variables destinadas al command parser
String inputString = ""; // string para almacenar los datos que entran
boolean stringComplete = false; // permanecerá en falso hasta que el string no haya
terminado
String cmd_disp_a, cmd_disp_t, cmd_on, cmd_off, ack,conectado ="";

//Inicialización de variables que rigen los estados el programa
boolean encendido = true;
boolean habilitado = false; //soft-start

//Declaración timer-interrupcion
void startTimer(Tc *tc, uint32_t channel, IRQn_Type irq, uint32_t frequency) {
  pmc_set_writeprotect(false);
  pmc_enable_periph_clk(ID_TC7); //cambiar esto para cada timer
  TC_Configure(tc, channel, TC_CMR_WAVE | TC_CMR_WAVSEL_UP_RC |
  TC_CMR_TCCLKS_TIMER_CLOCK1
  | TC_CMR_ACPA_CLEAR | // RA compare sets TIOA
  TC_CMR_BCPB_SET | // RB compare clear TIOB
  TC_CMR_ACPC_SET // RC cuando se llega al final
  | TC_CMR_ASWTRG_SET // set TIOA when software trigger
  | TC_CMR_BCPC_CLEAR // clear TIOB when RC
  | TC_CMR_BSWTRG_CLEAR // clear TIOB when software
  | TC_CMR_EEVT_XC0 // clear the EEVT para usar B as output
);

//Inizalización valor de registros
TC_SetRA(chTC, canal, 420); //10US
TC_SetRB(chTC, canal, 462); // 11US, 1US DE DEAD MINIMO
TC_SetRC(chTC, canal, rc);
TC_Start(chTC, canal);

// Habilitar la interrupción due al RC
tc->TC_CHANNEL[channel].TC_IER=TC_IER_CPBS;
// Deshabilitar otros tipos asociados al mismo
tc->TC_CHANNEL[channel].TC_IDR=~TC_IER_CPBS;
// Habilitar la interrupción
NVIC_EnableIRQ(irq);
}
void setup(){

  pinMode(24,OUTPUT);
  pinMode(26,OUTPUT);
  Serial.begin(9600);
```

Integración de control, medida y ensayos de un electroporador para tratamientos quirúrgicos

```
//Reserva de 200 bytes para string de entrada
inputString.reserve(200);
// Comandos disponibles
cmd_disp_t = String("DISP\n");
cmd_on = String("ON\n");
cmd_off = String("OFF\n");
cmd_dis= String("DESHABILITAR\n");
cmd_en= String("HABILITAR\n");
conectado=String("CONECTADO\n");

//Sacaremos el resultado de pulsos por los pines de la interrupcion, configuracion
completa del pin e inicializacion
// Habilitado como Puerto C.25 (PIN 3)
int result = PIO_Configure (PIOC,
PIO_PERIPH_B, //DUDA PIO_PERIPH_A
PIO_PC28B_TIOA7,
PIO_DEFAULT);
// Habilitado como Puerto C.26 (PIN 10)
int result2 = PIO_Configure (PIOC,
PIO_PERIPH_B,
PIO_PC29B_TIOB7,
PIO_DEFAULT);

PIOB->PIO_PER = PIO_PB17; // HABILITA EL B17 (P8) , COMO I/O
PIOB->PIO_OER = PIO_PB17; // SALIDA , ENTRADA?
PIOB->PIO_PUDR = PIO_PB17; // DESACTIVA PULL-UO RESISTOR INTERNOS

analogWriteResolution(10); // Modifica la salida analogica con 10 bit (1024 niveles)
analogReadResolution(10); // Modifica la entrada analogica con 10 bit (1024 niveles)

//Llama al timer
startTimer(chTC, canal, TC7_IRQn, frecuencia);

Serial.println("INICIALIZADO");
}
// parsing - commands

void DISP(){

int32_t ciclos_on = round(42*tiempo_on);
num_pulsos_count = num_pulsos; // contador de pulsos

Serial.print ("Pulsos enviados:");
Serial.println (num_pulsos);
Serial.print ("PULSO(us):");
Serial.println(ciclos_on);

TC_SetRA(chTC , canal, ciclos_on ); // actualizacion de salida, 340ns
TC_SetRB(chTC, canal, ciclos_on+42); // actualizacion de dato salida con lms
TC_Start(chTC, canal);
armado = false;
ack = "ACK";
}

void ON(){
encendido = true;
delay(2000); //soft-start
habilitado=true;
ack = "ACK";
digitalWrite (24,HIGH);
digitalWrite (26,LOW);
}

void OFF(){
encendido = false;
delay(2000); //soft-start
habilitado = false;
ack = "ACK";
digitalWrite (26,HIGH);
digitalWrite (24,LOW);
}

void PW(){
String dato_ent = "";
int x = 2;
```

```

do
{
dato_ent += inputString.charAt(x); // check the sensors
x++;
} while (inputString.charAt(x) != '\n');

x = atoi(dato_ent.c_str()); // libreria de conversi3n a enteros para char
pointer
if (x>100) {
x = 100;
}
if (x<10) {
x = 10;
}
tiempo_on = x;
Serial.println(x);
ack = "ACK";
}

void PN (){
String dato_ent = "";
int x = 2;
do
{
dato_ent += inputString.charAt(x); // check the sensors
x++;
} while (inputString.charAt(x) != '\n');
x = atoi(dato_ent.c_str()); // libreria de conversi3n a enteros para char pointer
if (x>500) {
x = 500; //maximo 500 pulsos soportaria el paciente
}
if (x<1) {
x = 1;
}
num_pulsos = x;
Serial.println(x);
ack = "ACK";
}
void CROWBAR(){
analogWrite(3,0);
analogWrite(10,0);
}

void loop(){

entrada_crowbar= analogRead(analogPin);

if (stringComplete) {
ack = "NACK";
if (inputString==conectado){
digitalWrite(28,HIGH);
}
if (inputString==cmd_on){
ON();
}
if (inputString==cmd_off){
OFF();
delay(2000); //crowbar
if(entrada_crowbar<=200){ //equivalente a 150v
CROWBAR();
}
}
if (inputString==cmd_disp_t){
if( encendido=true && habilitado=true){
DISP();
}
}
if (inputString==cmd_dis){
DISABLE();
}
if (inputString==cmd_en){
ENABLE();
}
if (inputString.startsWith("PW")){

```

Integración de control, medida y ensayos de un electroporador para tratamientos quirúrgicos

```
        if(habilitado=true && encendido=true){
            PW();
        }
    }
    if (inputString.startsWith("PN")){
        if(habilitado=true && encendido=true){
            PN();
        }
    }

    Serial.println(inputString);
    Serial.println(ack);

    // vaciamos lo almacenado en los string
    inputString = "";
    stringComplete = false;
}
}
void TC7_Handler()
{
    //Se acepta la interrupcion con TC_GetStatus
    TC_GetStatus(chTC, canal);

    PIOB->PIO_SODR = PIO_PB17; // modifica la salida del registro de acceso a PIO
    PIOB->PIO_CODR = PIO_PB17; // vacia el registro

    num_pulsos_count = num_pulsos_count - 1;
    if (num_pulsos_count < 1) {
        TC_Stop(chTC, canal); //finalizamos la interrupción
    }
}
void serialEvent()
{
    while (Serial.available()) {
        // get the new byte:
        char inChar = (char)Serial.read();
        Serial3.write(inChar);
        // add it to the inputString:
        inputString += inChar;

        if (inChar == '\n') {
            stringComplete = true;
        }
    }
}
```

ANEXO 4 : APP INVENTOR 2.

Se trata de un framework creado inicialmente por el MIT (Instituto Tecnológico de Massachusetts) y fue más tarde adquirido por Google con el fin proporcionar una herramienta que permita crear tu propia aplicación para dispositivos Android. App Inventor 2 es al mismo tiempo un lenguaje de programación, una herramienta de diseño y un entorno de desarrollo de aplicaciones.

El paso inicial, la construcción de interfaz de usuario en la herramienta **App Inventor Designer**. Permite el diseño de las pantallas de la aplicación añadiendo los componentes necesarios para su gobierno. Los componentes aparecen catalogados en categorías, estas son: *User Interface, Layout, Media, Drawing and Animation, Sensors, Social, Storage, Connectivity, LEGO MINDSTORMS*. En cada una de ellas, se encuentran los interesantes componentes a implementar.

La creación del código se lleva a cabo usando bloques de programación (**App Inventor Blocks**) Estos bloques se colocan para construir bucles, condiciones, variables, etc. que permiten pensar lógicamente y solucionar problemas de forma metódica.

Dichos bloques están clasificados mediante un criterio de colores que indica en parte su función o que componente al que hace referencia.

A continuación se detallan los posibles bloques a manejar y sus características.

- **Sentencias:** Cubren la necesidad que tiene el programador de decirle al programa que ejecute algo. Expresan una acción a realizar con un verbo imperativo. Los más habituales *call, set* y *def*. Estos bloques están coloreados en morado y verde.



- **Condiciones:** Permiten realizar acciones en función de que ocurra o no algo. Tendremos estructuras como *if-then, if-then-else, while* y *choose*. El color en el que representan es mostaza.



- **Bucles:** Provoca la ejecución repetida de varias sentencias. Se dispone de bucles como *while*, o *for each*.



While, ejecutara el bloque de sentencias encajado en do, mientras se cumpla la condición que se encaja en test. Foreach, permite iterar entre los elementos de una lista.

- **Variables:** Se proporcionan herramientas para definir y cambiar variables de tipo numérico o texto.



- **Eventos:** Será imprescindible cuando el usuario desee interactuar con la aplicación y su interfaz. Los eventos realizaran acciones en respuestas a los eventos que produzca el usuario. Se usarán eventos en la pulsación de botones por ejemplo.



Este entorno, nos aporta ventajas y facilidades por un lado, pero por otro encontramos ciertos hándicap que no podemos obviar.

Ventajas :

- Posibilidad de creación de aplicaciones sencillas y aptas para cualquier tipo de smartphone de sistema operativo Android.
- Software libre y multiplataforma, requiere únicamente de navegador y la máquina de Java instalada.
- Construcción de sentencias de forma univoca, se reduce la posibilidad de cometer un error de programación por el desarrollador.
- La forma de ensamblado de los algoritmos favorece el evitar dar lugar a errores.
- Visión de conjunto de programación de forma clara.

Inconvenientes:

- El entorno de desarrollo App Inventor 2 no cuenta con certificación SIL (Safety Integrity Level). [7]
- Rigidez de opciones de diseño, en cuanto a bloques de instrucciones y componentes para la interfaz de usuario.

ANEXO 5 : Variables, componentes y bloques de App Android.

VARIABLE/COMPONENTES	VALORES	DESCRIPCIÓN
NUMERO_PULSOS	1 a 500	Spinner. Cantidad de pulsos que tendrá la ráfaga de la señal
TIEMPO_ON	10 a 100	Spinner. Ciclo de trabajo de los pulsos de la señal
VISTA_MONITOR		TextBox. Emulador de Monitor Serie donde se visualizan los comandos
BluetoothClient1		Connectivity. Dispositivo bluetooth al que se conectará el usuario (módulo de Arduino)
NOTIFICACIÓN_CONECTADO	"EXISTE CONEXIÓN CON EL DISPOSITIVO BLUETOOTH"	Notifier. Informa al usuario.
HABILITADO /DESHABILITADO	"HABILITADO" "DESHABILITADO"	Notifier. Informa al usuario.
DESHABILITADO	"DESHABILITADO"	Notifier. Informa al usuario.
RELOJ_HAB	2 segundos	Clock. Permite cumplir seguridad soft-start
RELOJ_DESHAB	2 segundos	Clock. Permite cumplir seguridad soft-start y crow-bar.
ON/ OFF		Button. Encender/apagar app.
APLICAR_PULSOS/APLICAR_TIEMPO		Button. Modifican parámetro de la onda.
LIMPIAR		Button. Vacía VISTA_MONITOR.

Tabla 4 Descripción de variables y componentes

Integración de control, medida y ensayos de un electroporador para tratamientos quirúrgicos

COMPONENTE	When EVENTO	do	
CONECTAR_DISPOSITIVO	.BeforePicking	CONDICIÓN	If(BluetoothCliente1.Available)
			Then (SENTENCIA: set CONECTAR_DISPOSITIVO.Elementss)
CONECTAR_DISPOSITIVO	.AfterPicking	SENTENCIA	Call (BluetoothClient1.Conect)
		CONDICIÓN	If(BluetoothClient1.Available)
			Then(SENTENCIA: NOTIFICACIÓN_CONECTADO.ShowAlert)
ON	.Click	SENTENCIA	Set(VISTA_MONITOR.text) to ON
		CONDICIÓN	If(BluetoothClient1.Available)
			Then(SENTENCIA: call (BluetoothClient1.SendText) text ON
		SENTENCIA	Set(RELOJ_HAB.TimmeEnabled) to true
		SENTENCIA	Call(HABILITADO.ShowAlert)=HABILITADO
SENTENCIA	Set(RELOJ_HAB).TimerEnabled to false		
OFF	.Click	SENTENCIA	Set(VISTA_MONITOR.text) to OFF
		CONDICIÓN	If(BluetoothClient1.Available)
			Then(SENTENCIA: call (BluetoothClient1.SendText) text OFF
		SENTENCIA	Set(RELOJ_DESHAB.TimmeEnabled) to true
		SENTENCIA	Call(HABILITADO.ShowAlert) notice DESHABILITADO
SENTENCIA	Set(RELOJ_DESHAB).TimerEnabled to false		
APLICAR_PULSOS	.Click	SENTENCIA	Set(VISTA_MONITOR. Text) to NUMERO_PULSOS.selección
		CONDICIÓN	If(BluetoothClient1.Available)
	Then(call(BluetoothClient1.SendText) text PN, NUMERO_PULSOS.		
APLICAR_TIEMPOS	Click	SENTENCIA	Set(VISTA_MONITOR. Text) to TIEMPO_ON.selección
		CONDICIÓN	If(BluetoothClient1.Available)
	Then(call(BluetoothClient1.SendText) text PW, TIEMPO_ON.		
DISPARAR	Click	SENTENCIA	Call(blueetoothClient1.SendText) text DISPARO
		SENTENCIA	Set(VISTA_MONITOR.text) to DISPARO
LIMPIAR	.Click	SENTENCIA	Set(VISTA_MONITOR.text) to " "

Screen1	.Initialize	SENTENCIA	Set(RELOJ_HAB.TimerEnabled) to false
		SENTENCIA	Set(RELOJ_HAB.TimerEnabled) to false

Tabla 5 Descripción de métodos y bloques

ANEXO 6 : REFERENCIAS y BIBLIOGRAFÍA.

- [1] E. Maor, A. Ivorra, J. Leor, and B. Rubinsky, "Irreversible Electroporation Attenuates Neointimal Formation After Angioplasty," *Biomedical Engineering, IEEE Transactions on*, vol. 55, pp. 2268-2274, 2008.
- [2] A. Ivorra, "Tissue Electroporation as a Bioelectric Phenomenon: Basic Concepts" pp. 23-61 in "Irreversible Electroporation" Ed. B. Rubinsky, 2010, Springer-Verlag Berlin Heidelberg.
- [3] Edinger A. L., Thompson C. B. Death by design: apoptosis, necrosis and autophagy. *Curr. Opin. Cell Biol.* 2004.
- [4] Miller, Gary M. Modern electronic communication / Gary M. Miller, Jeffrey S. Beasley . - 7th ed. Upper Saddle River, New Jersey [etc.] : Prentice Hall, cop. 2002
- [5] <https://www.arduino.cc/en/Main/ArduinoBoardDue>
- [6] PFC: J. Iranzo, "Diseño e implementación inversor de alta potencia para sistema electrónico de tratamiento de cáncer mediante electroporación irreversible".
- [7] M. Charwood, S Turner and N. Worsell, UK Health and Safety Executive Research Report 216, "A methodology for the assignment of safety integrity levels (SILs) to safety-related control functions implemented by safety-related electrical, electronic and programmable electronic control systems of machines", 2004. ISBN 0 7176 2832 9